ROBUST PREDICTIVE MODELING OF RELATED GENE EXPRESSION DATA VIA MULTI-SOURCE TRANSFER RULE LEARNING

by

Henry Ato Ogoe

BSc, KNUST, Kumasi, Ghana, 2001

MSc, Åbo Akademi University, Turku, Finland, 2007

M.S., University of Pittsburgh, Pittsburgh, USA, 2013

Submitted to the Graduate Faculty of

the School of Medicine in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2016

UNIVERSITY OF PITTSBURGH

SCHOOL OF MEDICINE

This dissertation was presented

by

Henry Ato Ogoe

It was defended on

July 5, 2016

and approved by

Dr. Vanathi Gopalakrishnan, Associate Professor, Biomedical Informatics

Dr. Gregory F. Cooper, Professor, Biomedical Informatics

Dr. Xinghua Lu, Professor, Biomedical Informatics

Dr. Shyam Visweswaran, Associate Professor, Biomedical Informatics

Dissertation Director: Dr. Vanathi Gopalakrishnan, Associate Professor, Biomedical Informatics

ROBUST PREDICTIVE MODELING OF RELATED GENE EXPRESSION DATA VIA MULTI-SOURCE TRANSFER RULE LEARNING

Henry Ato Ogoe, PhD

University of Pittsburgh, 2016

The advent of high-throughput genomics has led to the accumulation of copious amounts of biomedical data such as gene expression, made available through public repositories like the NCBI's GEO. Meanwhile, the digitization of biomedical literature into repositories such as PubMed, have motivated the creation of curated knowledge bases like the Gene Ontology. Pooling information from such repositories and integrating it with predictive modeling of similar biomedical data from multiple studies, could lead to models that are more robust. Most current methods are unable to leverage background knowledge, referred to herein as catastrophic forgetting, and often produce black-box models that are difficult for humans to interpret.

In this era of precision medicine, there is thus a critical need for effective methods that could incorporate background knowledge from multiple sources, and yet produce simple to understand models from biomedical datasets. This dissertation develops four novel frameworks: (i) TRL-FM, (ii) KARL, (iii) MS-TRL, and (iv) iTRL, which use transfer rule learning to incorporate background knowledge from multiple sources for predictive modeling of gene expression datasets. They provide significant extensions to an existing method, TRL that leveraged background knowledge from single sources. This work tests the hypothesis that "incorporating background knowledge from multiple sources into predictive modeling via the transfer rule learning approach leads to models that contain more robust propositional rule patterns than learning without any background knowledge or just from a single source."

To test this hypothesis, I compared the accuracy and coverage of predictive models that

were produced with the methods developed herein, to the baseline models, using 25 gene expression datasets from 5 studies of brain, breast, colon, lung, and prostate cancers. The results showed that the former, produce on average, statistically significantly more robust models than the latter. Also, KARL, MS-TRL, and iTRL provide mechanisms that could be used to discover both domain-specific and domain-independent robust rule patterns.

The methods developed herein augment extant capabilities of predictive modeling techniques to utilize and build robust, easy-to-interpret rule models from sparse, single, diverse sources of biomedical data and knowledge. These methods can be easily extended to other application domains beyond biomedicine.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.0 INTRODUCTION

In recent years, advances in biomedical research have led to an explosion of data that provides several avenues for improved knowledge discovery, data mining, and biomedical decision making. With the advent of high-throughput techniques, for instance, several 'omic' data, like the microarray, that describe measured biomarkers in bodily fluids or tissue are accumulating at a fast pace. What is more, the adoption of standards and reporting requirement guidelines such as Minimum Information About a Microarray Experiment (MIAME) [1] as well as the establishment of public repositories for microarray data such as the Gene Expression Omnibus (GEO) [2] and the ArrayExpress [3], have made it possible for gene expression data resulting from related studies to be reused and shared. As of the time of writing this manuscript, ArrayExpress and GEO alone have a combined data for more than 100,000 studies with over three million assays. Meanwhile, the digitization of key biomedical research findings and publications into publicly available repositories, like PubMed, have motivated the creation of curated knowledge bases such as the Gene Ontology (GO) [4], Kyoto Encyclopedia of Genes and Genome (KEGG) [5], or Ingenuity ® Knowledge Base (IKB), just to name a few. There is thus an explosion of biomedical data and knowledge that when effectively harnessed could lead to significant progress towards the goal of precision medicine.

Precision medicine is the use of 'omic' and other relevant data to describe tailored and accurate medical treatments selected according to individual characteristics of a patient. Thus, to effectively pool information from as many sources as useful in order to improve accuracy and precision of disease diagnosis, prognosis, and/or treatment. According to the NRC's report on precision medicine, however, there is a growing shortfall of better tools and mechanism to commensurate the data explosion [6].

To effectively automate knowledge discovery and learning in the space of data is an active

research endeavor in biomedical science and medical decision making. Predictive modeling refers to a collection of machine learning methods which aim at estimating a mathematical relationship between a target variable (e.g., a disease state) and several predictor variables. The goal is to make inference with the estimated relationship on the state of a newly unseen individual, given its measured predictor variable values. With ample data predictive modeling could be used for critical medical decision making like disease diagnosis, prognosis, and/or treatment. Supervised machine learning methods, for instance, have been applied successfully on gene expression profiles to predict cancer diagnosis [7] and prognosis [8]. Thus, in the context of precision medicine, combining information from multiple related studies for predictive modeling could lead to the discovery of new biological insight as well as better models for a more effective medical decision making, in light of the data explosion.

1.1 THE PROBLEM

With the plethora of biomedical data and knowledge repositories, integrating information from related studies designed to study the same/similar biological problem could lead to the discovery of predictive models that are more generalizable, reliable, and precise. Inherent nuances within and between such related studies and models, however, make predictive modeling of related biomedical data, such as gene expression datasets, a non-trivial task. Below are some of the challenges:

1. **Curse of dimensionality**

   High-throughput techniques for generating gene expression data are both a blessing and a curse. They are a blessing because thousands of biomarkers can be assayed simultaneously, ensuring high efficiency for estimating expression levels of biomarkers. They are a curse because most of the thousands—to tens of thousands—measured biomarkers are irrelevant for disease state classification. In addition, traditional data mining algorithms were not intended for such high dimensional data. Developing a reliable predictive model with them is a major challenge in machine learning, because of the "over-fitting" problem [9].

2. **Curse of dataset scarcity**

   Gene expression datasets, like most 'omic' data, are fraught with small sample sizes, usually in the region of tens or hundreds. Studies [10] have shown that for predictive models learned on such data to command high stability, generalization power, and reliability, a large sample size—thousands—is required. Unfortunately, due to cost and limited tissue (or biofluids) availability, it is prohibitive to obtain 'omic' data with such large sample size.

3. **Variability & noise**

   In the design of high-throughput experiments, three types of variations can occur in the final observed data: biological variations, technical variations, and treatment effect [11, 12]. Biological variations arise due to heterogeneity among individuals (i.e., sample population), tissues, and/or environmental factors. Technical variations, on the other hand, refer to all kinds of experimental variability or artifacts introduced when two identical samples are assayed and analyzed with different equipment, protocols, and/or methods. Treatment effect refers to the difference between two experimental groups (e.g., case versus control) depending on the underlying experimental hypothesis. Because of these variations, it is not effective to combine, naively, two or more related datasets in order to boost sample size.

4. **Lack of transferability**

   In most cases, a highly predictive biomarker generated from one study can suffer a marked decrease in performance when tested on another related study [10]. Results from a knowledge discovery process on related datasets can be different. For instance, several microarray studies that have attempted to address similar prediction tasks have reported sets of predictive biomarkers, which are either entirely different or show very little overlap [10, 13, 14]. The ability to discern such subtleties are essential for model performance, however, most predictive modeling techniques are unable to identify, incorporate, nor transfer background information from related studies into new models; a phenomenon that is also known as *catastrophic forgetting* [15]. This lack of transferability, which is mostly caused by inherent heterogeneity within diseases (e.g., cancer), can affect negatively, the discovery of reliable biomarkers and predictive patterns that are useful for

verification studies between independent, but related, 'omic' datasets.

5. **Model interpretability**

   Most of the statistical learning and data mining methods for analyzing high-dimensional data, like microarrays, generate predictive models that are difficult to interpret by humans, even though they can potentially yield high accuracies. Striking a balance between model interpretability and predictive performance can be a daunting task for the biomedical scientist.

1.2 CURRENT METHODS

Methods for predictive modeling can be roughly categorized into two main groups. Given an 'omic' data like gene expression, the first group, statistical or pattern-recognition learning techniques like logistic regression, k-nearest neighbor, artificial neural networks (ANN), or support vector machines (SVM) [16] could be used to approximate a mathematical function for future predictions. The second group involves inductive learning of symbolic descriptions, such as decision trees [17], classification rules [18], or logical representations [19]. Although the former can result in models with relatively high predictive performance (e.g., SVM), majority of them (e.g., ANN, SVM) suffer from the interpretability problem. This work therefore focused on the latter group of methods, particular RL [18], which has been used extensively, over three decades, for predictive rule modeling of 'omic' datasets.

Given a list of training data examples, the goal of a symbolic learning algorithm, like RL, is to identify a set of classification rules, i.e., a *rule model*, which can be used to predict new data instances. A data example, herein, refers to a set of variable-value pairs, where a variable (e.g., a gene) is a place holder for an object of a domain/world. Meanwhile, a data instance refers to an example without a class label, while a dataset denotes a set of examples. An instance is *covered* by a rule if it logically satisfies the rule's condition, while *coverage* is the fraction of examples that are covered by a rule model. Furthermore, a rule model is *complete* and *consistent* if it covers and accurately predicts the class labels of all data examples, respectively. Herein, a rule model is said to be *robust* if it is complete

and consistent. This definition of robustness would be used throughout this document as a guiding principle to gauge two or more rule models.

RL and several other symbolic methods can adequately address the model interpretability problem, however they do suffer from the data scarcity as well as transferability problem, which make them less robust. Consider, for example, two related microarray studies on lung cancer, where one involves patient cohorts from Bostom, while the other from Pittsburgh. While vital information contained within these related studies could be leveraged to improve predictive performance of models learned on a dataset from each study, RL and most single-source based algorithms cannot incorporate such background knowledge into their model development. That is *catastrophic forgetting*.

To address some of these challenges several methods have been proposed for combining information from multiple and independent microarray studies that were designed for the same biological problem (e.g., survival of prostate cancer)—in order to discover more generalizable models by boosting sample size. Majority of these methods can be categorized under two main approaches: meta-analysis and analysis by data merging [20–22]. Meta-analysis methods combine results of individual studies (e.g., classification accuracies, ranks, p-values, etc.) at the inference level. By contrast, merging methods, like batch effect removal techniques, after transforming the expression values from different studies into numerically comparable measures, integrates microarray data at the expression value level. Furthermore, the output of these methods could be fed into machine learning algorithms to develop classification models.

The major limitations about these methods are that they do not address the transferability and variability issues well. Meta-analysis relies on statistical significance to pool information from different studies to draw inferences. As alluded to above, however, a biomarker that is statistically significantly predictive in one study can manifest a different behavior in another study, which is designed to solve the same biological problem. In addition, most of these methods are unable to incorporate expert and domain-specific knowledge to guide model construction and interpretation, nor transfer information from one dataset to another in order to boost reliability of the integrative analysis.

To address these challenges, Ganchev and colleagues proposed a novel framework (an

extension of RL), called transfer rule learning (TRL), which leveraged the concept of transfer learning to build integrative, modular, and interpretable predictive rule models from two datasets [23]. Transfer learning (TL) is The ability of a system to recognize and adapt knowledge, and skills learned in previous domains/tasks to novel domains/tasks, which share some **COMMONALITY** [24]. Given amble background knowledge, humans are able to learn efficiently and make better informed decisions using the mechanism of TL.

Given two 'transcriptomic' datasets of related studies (see example above), where one is designated as the source and the other as target, TRL builds predictive rule models, while using the concept of TL. First, it learns a rule model on the source, and second, it transfers knowledge learned from the source model to seed learning of a new predictive rule model on the target. By this mechanism, TRL addresses the data scarcity, as well as the transferability problem inherent in RL and other single-source machine learning algorithms.

In the spirit of precision medicine, however, TRL has limited capabilities. First, while gathering information from multiple sources for transfer might lead to better learning and classification performance on the target, the current implementation of the TRL framework cannot pool rules learned from multiple data sources for transfer (i.e., it can only transfer knowledge between a single source and a single target). Second, it cannot imbibe relevant biological knowledge from external sources to augment the knowledge discovery process. Extracting, combining, and abstracting vital information from domain experts, literature, or domain knowledge bases to augment transfer learning, could be an invaluable proposition for developing robust predictive rule models. Third, it has its own flavor of *catastrophic forgetting*. While transferring background knowledge, in the form of rules, from the source to target, it fails to account for the predictive performance of individual rules in previous studies. As alluded to above (see section 1.1), some biomarkers and rule patterns may perform well in one study, but not another. Taking cognizance of this phenomenon, while transferring knowledge between the source and target could potentially improve model robustness. Last, for knowledge transfer to be effective and meaningful it is essential to capture the relatedness between the source and target datasets. TRL's mechanism of establishing this relatedness is to identify overlapping variables between the source and target. However, studies have shown that different classification models built on independent, but related, microarray

datasets can contain different sets of biomarkers with little overlap. In addition, models based on different variable sets can yield similar classification performance when tested on the same validation dataset [25, 26]. This means that TRL's naive approach for establishing relatedness might not inure well to the benefit of knowledge transfer. In humans, for example, the *TP53* gene, which encodes the tumor protein *p53*, is known to play a key role in the activation and/or control of apoptosis. Meanwhile, caspase-6, an effector caspase, which is encoded by the *CASP6* genes, cleaves to other proteins to trigger the apoptosis process [27]. Superficially, *TP53* and *CASP6* are different, but they both play a prominent role in apoptosis. Assuming these genes are significantly predictive in the source and target datasets respectively, TRL, including many other integrative methods, cannot leverage their commonality to facilitate knowledge transfer because they are not identical. Thus hampering, potentially, the robustness of predictive models that are developed by them.

Considering the challenges and limited capabilities of the current methods, one may ask, "in the wake of biomedical data and knowledge explosion and the drive for precision medicine, how do we effectively extract and combine background knowledge from multiple sources to transfer and learn simple, but, robust predictive models on a target dataset?" Thus, there is a critical need for a new approach to harness the vast amounts of information contained in the burgeoning biomedical data and knowledge repositories in order to provide alternate, but more effective tools, to champion precision medicine. To that end, this dissertation presents a new approach, based on four different, but related algorithms.

1.3 THE APPROACH

This dissertation explores four novel frameworks: (i) Transfer Rule Learning via Functional Mapping (TRL-FM), (ii) Knowledge Augmented Rule Learning (KARL), (iii) Multiple Source Transfer Rule Learning (MS-TRL), and (iv) Incremental Transfer Rule Learning (iTRL), which offer substantial extensions to TRL [28] by addressing the above-highlighted limitations, so that predictive rule modeling, via transfer learning, could be more robust. Figure 1 illustrates an overarching scheme that encapsulates the proposed extensions devel-

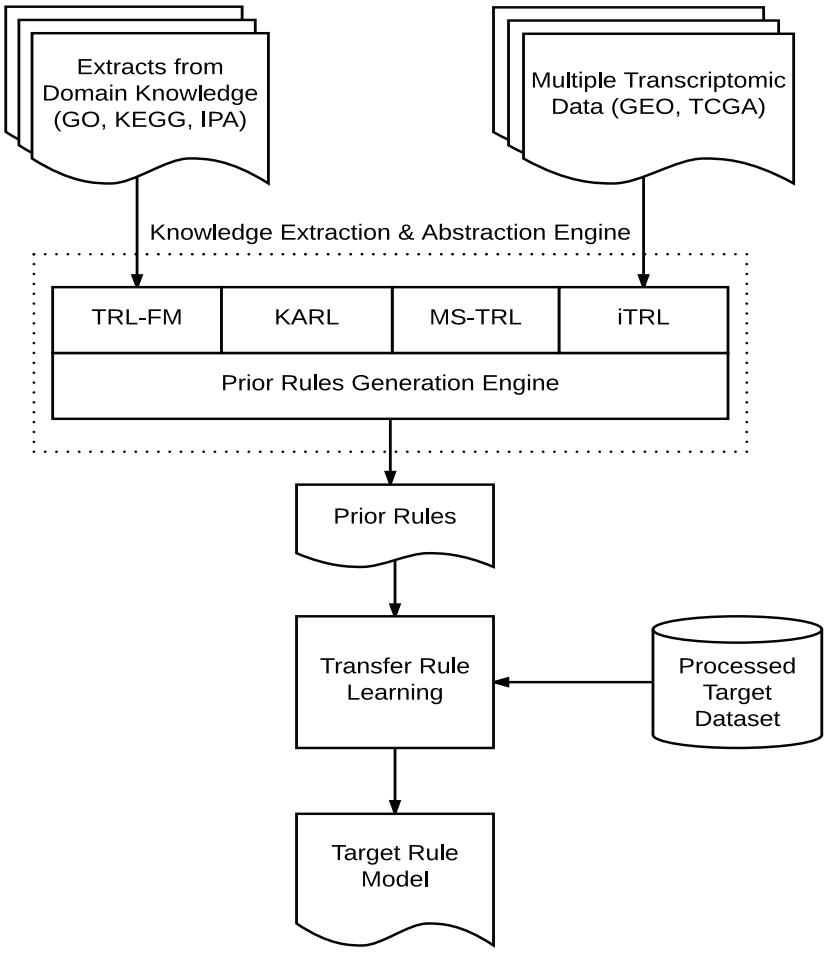

Figure 1: A simplified scheme that encapsulates the multi-source transfer rule learning frameworks developed herein. First, pertinent knowledge are extracted from two distinct sources: (1) domain knowledge bases (e.g., GO, KEGG, IPA), and (2) multiple related transcriptomic data sets (e.g., GEO, TCGA). Second, the knowledge extraction engine, which involves a suite of frameworks (TRL-FM, KARL, MS-TRL, and iTRL), implements diverse procedures to process the multi-source background into simplified structures for the prior rules generation engine. Third, the prior rules generation engine converts the abstracted knowledge into a list of prior rules. Last, using transfer rule learning (see Figure 30), information from the prior rules are combined with the target dataset to develop a new target rule model.

oped herein. Its main goal is to harness vital information contained in the vast amounts of biomedical data (and knowledge) resources to augment predictive rule modeling via the transfer learning paradigm.

The main contribution of this framework is the Knowledge Extraction & Abstraction Engine (KEAE), which consists of two main parts. The first, i.e., the knowledge extraction engine, consists of four sub-components that implement diverse sub-routines to extract information from multiple sources to facilitate the induction of a predictive rule model on a target dataset. The second, a Prior Rules Generation Engine (PRGE) complements the first by abstracting the source (or background) knowledge into a prior hypothesis in the form of classification rules. The PRGE is a polymorphic sub-routine in that each of the four knowledge extractions engines abstracts and generates prior rules differently. The four components of the KEAE, i.e., TRL-FM, KARL, MS-TRL, and iTRL, including their nuances, are briefly explained as below:

TRL-FM The purpose of TRL-FM is to lax TRL's requirement for establishing commonality between the source and target in order to transfer knowledge—that is, identical variables must occur. It extracts vital domain knowledge from a biomedical knowledge repository (i.e., the Gene Ontology) into an ontology-based functional modules (FMs)—a group of variables that perform the same/similar functions. The FMs abstracts commonalities between source and target variables, and thus could be used to map even non-identical, but common variables, to facilitate prior rules generation. This mechanism improves completeness, and hence robustness of a TRL-FM rule model, given the same target (see example in section 1.2).

KARL The purpose of KARL is to extract and abstract germane domain knowledge from multiple experts, literature, and/or knowledge bases, to directly augment search for robust predictive rules. It addresses TRL's inability to imbibe relevant biological knowledge from external sources to augment the knowledge discovery process. In addition, it affords a user the flexibility to incorporate subjective domain information into the knowledge discovery process of rule learning. For instance, it provides a generic framework such that vital information from a domain (e.g., characteristics of cancer or schizophrenia) could be used as a guide to discover interesting rule patterns. The rule patterns

`IF VEGFA=Low ==> CONTROL` and `IF VEGFA=Up ==> CANCER`, for example, could be given a relatively high priority during search for propositional rules because a highly-expressed `VEGFA` can be associated with proliferation of the cell—a hallmark of cancer. Using a biomedical knowledge repository like Ingenuity® Knowledge Base, KARL's KEAE first extracts desirable domain knowledge, which is subsequently abstracted and encoded into a domain-specific data structure—a functional lookup table. Information encoded in the lookup table augments the generation of prior rules for transfer learning.

MS-TRL MS-TRL is also aimed at addressing the single source limitation of TRL. Here, the notion of *multiple sources* is characterized by multiple related rule models, as opposed to knowledge bases àla TRL-FM/KARL fashion. What is more, MS-TRL alleviates TRL's version of catastrophic forgetting by implementing intelligent mechanisms for data transformations, as well as *remembering* and incorporating the predictive performances of rule patterns from related models into new ones. Consider, for instance, four related microarray studies on breast cancer, which involved patients from Boston, Pittsburgh, Stanford, and Michigan, respectively. Unlike TRL, MS-TRL is able to extract and abstract vital background knowledge from any three to seed learning of a new predictive model on the fourth. The ability to imbibe information from multiple models makes MS-TRL relatively more robust.

iTRL Like MS-TRL, iTRL can address the single-source limitation of TRL as well. The main difference between the two is that for the former, multiple related models are merged and processed into prior rules, while with the latter, prior rules are generated and updated from each source, one at a time. Thus, iTRL provides capabilities for TRL to be used for on-line learning, which could be particularly useful when all source datasets are not available at the same time.

As fig. 1 illustrates, the knowledge extraction and abstraction engine outputs a set of prior rules. Using the prior rules as a seed, the framework proceeds in a TRL fashion to induce a new predictive rule model on the target data. Thus, with this new approach we are now able to pool background knowledge from a wider and more informative space to augment transfer rule learning.

1.4 AIMS OF THE DISSERTATION

In the wake of biomedical data (and knowledge) explosion; the drive for precision medicine; the limitations of current methods; and hence the critical need for more effective methods, the aim of this dissertation is threefold. First, to present a novel approach that is founded on four different, but related, frameworks that are able to harness and combine background knowledge during search for predictive rule models from multiple sources of related gene expression data and biomedical knowledge bases using transfer learning. Second, to evaluate and compare the performance (i.e., robustness) of the frameworks developed herein with previous methods, using 25 real and publicly available gene expression datasets, which represent five respective studies on brain, breast, colon, lung, and prostate cancers. Last, to postulate that:

> *The multi-source transfer rule learning frameworks developed herein, TRL-FM, KARL, MS-TRL, and iTRL, produce on average more robust predictive rule models than those produced from a single-source transfer, or no transfer.*

1.5 SIGNIFICANCE

To the best of my knowledge, this is the first transfer learning study that provides diverse mechanisms to extract, abstract, and combine information from multiple-related biomedical data sources to develop robust predictive rule models. The significance of the frameworks developed herein can be viewed from two perspectives: (1) from an informatics point of view and (2) a biomedical viewpoint.

From an informatics viewpoint, the algorithmic extensions that were developed will improve classification rule learning with TRL and its predecessor, RL [18]. RL performs comparably or better than many other machine learning algorithms on several biomarker data mining case studies [29–32]. TRL was the first method to apply the concept of transfer rule learning for integrative biomarker discovery. Most of transfer learning methods are unsuitable for high-dimensional datasets, like biomedical datasets, because they generate models that use a large variable set and are difficult to interpret [28, 33, 34]. Like RL and TRL, the

extensions that this work proposes output rule models that are modular, intelligible, and well suited for knowledge discovery and predictive modeling tasks for biomedical datasets.

The algorithmic extensions developed herein will improve transfer rule learning in several ways. First, by pooling information from multiple source datasets, as opposed to one, to seed learning on a target dataset, it will more likely improve predictive performance, reliability, statistical power, and the identification of robust patterns. This is akin to receiving "advice" from multiple experts. As opposed to just one, whose advice may be good or bad; here you have more options to sift through the best (and worst) in order to make a better-informed decision. Most genomic data are fraught with noise and can lead to unreliable models. Combining information from multiple-related sources, however, could smoothen out inherent noise from each source, and provide a robust picture of the underlying knowledge. Second, by leveraging the notion of functional modules to capture and abstract biological commonalities among variables it will be more suited to tackle the transferability challenge, which current methods, including TRL are unable to address well. For instance, a biomarker might be statistically predictive in one study, but not another. FMs can be used as a pivot to map surrogate biomarkers between independent, but related 'omic' datasets. In addition, the FM mapping and knowledge augmentation (e.g., with KARL) mechanisms provides another source of information to enrich model development. The advantage these mechanisms have over other multi-source approaches, such as ensemble methods, is that, they are able to incorporate, explicitly, prior domain-information to augment the knowledge discovery process. Finally, the proposed algorithmic extensions provide generic frameworks that could be customized for cross-domain studies, like 'panomic' studies—a key driver for precision medicine.

From a biomedical standpoint, the frameworks put forward herein could be an invaluable tool set for translational scientists who are interested in precision medicine. In the advent of biomedical data explosion, these frameworks provide novel mechanisms that could discover robust patterns of distinct biomedical knowledge nuggets within and across homogeneous and heterogeneous datasets, respectively. Identifying similar and robust patterns across related disease types could have significant implications for treatment strategies. Various molecular profiling studies, for instance, have revealed that cancers from the same tissue/organ are

oftentimes distinct, while cancers of different tissue/organ can share common features [35,36]. Certain types of lung and head-and-neck cancers, for example, have similar features as some types of bladder [35] cancer. This means that by identifying these similar features via the proposed frameworks herein, could yield several efficient therapeutic options. An oncologist, for instance, could apply knowledge gained from treating squamous cell lung cancer to some bladder cancer cases that share the same characteristics. In sections 5.3.8, 5.4.4 and 5.5.3, the feasibility of KARL, MS-TRL, or iTRL for handling such use case are illustrated via experimental results, which are supported by literature-based evidence.

Thus, the multi-source transfer rule-learning frameworks developed herein could make meaningful contribution towards the precision medicine initiative. Given multiple related datasets, they could be used to discover simple, interpretable, and more robust rule models. In addition, they are capable of identifying patterns of domain-specific (i.e., distinct features) and domain-independent (i.e., common features) patterns within a specified domain.

1.6 OVERVIEW OF DISSERTATION

In chapter 2, I will highlight relevant background literature on the proposed frameworks. Some of the germane topics to be discussed are gene expression, analysis of microarray gene expression data, integrative analysis of gene expression data, transfer learning, and the foundation of rule learning. Chapter 3 describes, in details, the conceptual underpinnings, including implementation algorithms for all the multi-source transfer rule-learning frameworks developed and tested in this dissertation. Chapter 4, subsequently describes the design and experiments that were performed to ascertain the feasibility of the frameworks to address some of the problems highlighted in section 1.1. In chapter 5, I present results for the experiments, including a detailed analysis of them. Meanwhile, all additional materials (e.g., Supplementary results, and user manual for the implementation toolkit) have been provided in appendices A to F. Chapter 6 concludes the dissertation with highlights of the insights we gained from this work, the contributions it provide to knowledge within the biomedical informatics community, identified limitations, and proposed future work.

2.0 BACKGROUND

This chapter discusses related background literature that is relevant for predictive modeling of multiple gene expression data sets using transfer learning, the object of this work. Section 2.1 provides an overview of gene expression data and its utility for unraveling the underlying mechanisms of disease states, while section 2.2 highlights, in general, the computational methods that are used for the analysis of gene expression data to gain more biological insight. Section 2.3 examines, compares, and contrasts integrative data and/or modeling approaches (e.g., data merging, meta-analysis, ensemble learning) that have been developed to improve learning by combining information from multiple sources, the context of this work. Section 2.4 provides a brief survey of transfer learning, the overarching concept underpinning this work, while section 2.5 describes the foundations of rule learning, which are the building blocks for the methods put forward in this thesis.

2.1 GENE EXPRESSION

The abundance of mRNA transcript at particular time points can give an indication of the functional role of a gene, the underlying mechanism of a disease, or a potential drug response [37–39]. Given a biological sample (i.e., bodily fluid or tissue), gene expression profiling can be used to measure the abundance of each RNA transcript in the transcriptome—a collection of all *RNA* transcripts, including both protein coding *mRNA* and non-coding RNAs [38,40]. The technology for gene expression profiling has evolved over the past few decades. It started with the Southern blot [41], and its variant technique, the Northern blot, and then followed by the quantitative RT-PCR [42]. These pioneering technologies can only interrogate a

handful of genes at a time.

In the past two decades, the advent of high-throughput techniques such as microarrays (e.g., *cDNA* array or whole genome tiling array [38,43]) enabled gene expression profiling of, potentially, the whole transcriptome. Recently, the application of next generation sequencing technologies [44], like RNA sequencing (RNA-Seq) [45], to profile the transcriptome is becoming more popular because of its unprecedented accuracy and sensitivity [45,46]. Several studies in the literature have compared and contrasted the strengths and limitations of DNA microarrays and and RNA-Seq [47–49]. For instance, reports from some studies claim that the correlation of technical replicates of the two methods can be higher than 0.9 [50,51]. Meanwhile, RNA-seq has a wider dynamic range, and it is more sensitive than the microarray [47]. The improved sensitivity can enable it to identify more genes, while the wider dynamic range may increase accuracy [47,49]. However, the RNA-seq is more costly (about 3-5 times the cost of microarray per sample); it can generate very large files (about 30-40 times larger than the microarray); it requires extensive bioinformatics skills and computer resources; and since it is a new technology, there are lot of tools for analysis, yet no standard protocol to guide them [47].

This work focused on microarray gene expression since it is still the common choice for transcriptomic profiling. About 80% of gene expression data that are readily available in public repositories are microarrays [52]. In addition, the frameworks put forward are flexible such that they can be easily modified and equally applied to a wide spectrum of 'omic' datasets such as RNA-Seq, DNA methylation, etc., when available.

2.2 GENE EXPRESSION DATA ANALYSIS

The simple assumption underpinning microarray data analysis is that: "given that genes are expressed by transcribing into *mRNA*—which will be later used to synthesize proteins—if we are able to identify which (and how much) *mRNA* is present we should be able to determine which genes are expressed, including their corresponding expression intensity." That is, the number of *mRNA* molecules resulting from the transcription of a given gene can be

used to approximate its level of expression. Even though this assumption is not foolproof, it is the guiding principle for the design and analysis of almost every microarray experiment. Depending on the microarray technology (e.g., *cDNA* microarrays or oligonucleotide chips), gene expression intensities are quantified by means of fluorescence intensities that are captured by scanners into images. The images are subsequently turned into numbers, which forms the basis for further statistical analysis [39, 53]. Generally, microarray data analysis adhere to the following major steps: experimental design, preprocessing, and inference.

Experimental design involves the development of a plan to get the best out of the information being measured in order to answer the biological question adequately. For instance, ensuring reasonable sample replicates and sample size can reduce and increase variability and statistically power, respectively. After the expression intensity levels have been converted from images to their numerical equivalent, the data is preprocessed in order to ensure quality, harmonization, and interpretability of the expression values.

The preprocessing step can involve quality control (QC), normalization, data transformation, and data filtering. The goal of the QC process is to ensure that the transformed expression intensities are reliable. Several methods for microarray QC [54, 55] have been proposed, however, there is no specific standard adopted by the microarray community.

Normalization is also another essential preprocessing step. Its goal is to remove systematic variations and artifacts among different microarray experiments and platforms. It also allows consistency and harmonization of the data to ease comparison among different microarray experiments. Several approaches for normalization have been proposed and well discussed in literature [56–59]. In addition to normalization, other important preprocessing steps are data transformation and filtering.

Data transformation [60] involves the application of a specific function to change the data into a different form (e.g., log_2, or Z transform) while filtering can be used to simplify the data analysis by removing expression intensities with relatively low signals [61]. The output of these data preparation steps is a gene expression matrix, where the rows (tens of thousands) and columns (from 2 to hundreds) represent genes and samples, respectively. Depending on the type of microarray technology used, the values of the matrix denote gene expression (i.e., relative expression for two-channel array technologies and absolute

expression for single-channel technologies).

Finally, the output of the preprocessing step, the expression matrix, serves as input to various statistical analyses/hypotheses tests—the inference step—in order to answer the biological question for which the experiment was designed. Microarray experiments, in general, are designed with several inference objectives in mind. Majority of these inference objectives can be categorized into four groups namely, class comparison (i.e., differential gene expression or candidate marker identification), pathway analysis or functional enrichment, class discovery, and class prediction [62]. The next subsections review, briefly, the general inference approaches that are used to meet the objectives for each category.

2.2.1 Class Comparison

The class comparison question, which is also known as "difference in gene expression", seeks to identify genes whose expression levels are significantly different between two conditions. Given two class conditions, say primary lung tumor (condition A) and normal lung tissue (condition B), an example hypothesis is "there is a difference in gene expression between tissues belonging to condition A and B in the general population of lung tissues." Statistically tests like the t-test or ANOVA, for example, can be used to determine the significant difference between gene expression means of two or more groups. A comparative review of statistical methods for discovering differentially expressed genes from microarray experiments can be found from [63, 64].

2.2.2 Pathway Analysis

Generally, differential gene expression analyses result into a long list of genes. With this list, biomedical scientists are interested in associating the genes to functional classes in order to give them biological interpretation. Common approaches to seek biological interpretation is to associate the genes to named functional classes (also known as functional enrichment) contained in functional annotation databases like Kyoto Encyclopedia of Genes and Genomes (KEGG) [5] and the Gene Ontology (GO) [65]. Several methods, including their strengths and demerits, for implementing this approach have been proposed [66–68].

Two of the most commonly used functional enrichment methods are Gene Enrichment Analysis (GE) and Gene Set Enrichment Analysis (GSEA). Given a list of genes, the GE-based methods seek to identify whether a given term, say a KEGG pathway or a GO term appears more frequently (i.e., *enriched*) or less frequently (i.e., *impoverished*) in the given list than the population from where the genes are obtained. Population here can refer to a list of differentially expressed genes, set of genes captured by the microarray, or the genome. The hypergeometric test is usually used to determine the significance of any identified enrichment. In addition to the gene list, the GSEA [69] methods also require a numerical variable (usually the p-value) to rank the given gene list. Beginning with the ranked list a cumulative enrichment score based on the absence or presence of each gene from a known gene set is computed. To determine whether the known gene set is over-represented at the top or bottom of the list, a Kolmogorov-Smirnov statistic is used to compare the distributions of scores between the known gene sets and given gene list. A comparative analysis of these methods can also be found from [66, 68].

In addition to functional enrichment analysis, there are other approaches—biological network analysis—that can be used to identify functional groups or common disease associations among differentially expressed genes [70]. Majority of these methods, including *functional motifs discovery* [71] and *functional modules discovery* [72] from protein-protein interaction networks have been reviewed and/or discussed elsewhere [73].

2.2.3 Class Discovery

Apart from pathway and enrichment analysis of the gene expression matrix, class discovery, also known as clustering, is another essential approach that can be used to identify and group genes that exhibit similar expression patterns. Subsequently, the identified groups can be correlated to biological information—from pathway or functional annotation databases—to make inferences. It can be hypothesized, for instance, that functionally related genes are co-expressed (i.e., up or down regulated simultaneously) and therefore can be used as a basis for clustering.

Algorithms for cluster analysis of gene expression data abound [16]. Given a similarity

metric (say, expression pattern of genes across tissues) and the expression matrix as inputs, these algorithms can group genes, and even tissue samples, into desirable classes. When applied to genes, for instance, they can identify co-regulated genes or spatio-temporal expressions. Meanwhile, when applied to tissue samples, methods like hierarchical clustering can identify biological classes (e.g., tumor sub-types) or even experimental artifacts [74].

In spite of its wide usage for exploring the gene expression matrix, there are a couple of open problems in cluster analysis. Although several cluster analysis-based methods abound, there are no clear guidelines or consensus on, for instance, which particular metric to use to quantify the similarity among objects; how to determine the optimal cluster size; or how to validate produced clusters [75–79].

2.2.4 Class Prediction

Given an expression matrix, the goal of class prediction analysis is to develop a multivariate function or rule that can predict, accurately, the class membership (e.g., primary tumor or normal) of a new tissue sample based on its gene expression values. Assume that each sample of the array is labeled with a class, say $Y \in 1, 2, ..., J$, where J is the number of classes, and characterized by a vector of features, $X = (X_1, X_2, ..., X_G)$, which represent the expression values of G genes. The goal is to predict the Y value of a newly unclassified sample given its X.

Depending on the underlying biological question, class prediction functions can be divided into two main categories: *diagnostic* and *prognostic*. With the former, the developed function assigns a new sample to an existing category or disease. For example, Golub *et al.* [7] developed a classification model that could assign tumors to their respective subtypes. The goal of the latter is to predict the progress of patient's disease. Van't Veer *et al.* [8], for example, developed a predictive model that could determine whether a tumor may metastasize after given period.

Class prediction is an active research area in biomedical informatics, where a myriad of machine learning algorithms have been proposed or applied to microarray data [16]. Reviews [80, 81] have demonstrated that relatively simple and well-known machine learning methods

like Naive Bayes (NB), K-Nearest Neighbor (KNN), Linear Discriminant Analysis (LDA), Support Vector Machines (SVM), and decision trees (C4.5), perform well on most class-prediction tasks for microarray data. However, pertinent challenges persist. First, given an expression matrix, different methods may yield different classification models, which are not unique and produce different error rates. Thus, choosing an appropriate method for a particular class prediction task can be a challenging. This challenge can be tackled with some ensemble or model averaging methods [82, 83], even though they come with inherent challenges [84].

Meanwhile, interpretation of a class prediction model is very important. Unfortunately, majority of traditional machine learning algorithms are "black boxes". They might yield a high predictive performance but difficult to interpret. A careful balance of model interpretability and predictive performance is key for precision medicine.

In addition, as alluded to earlier, the gene expression matrix is characterized by the curses of dimensionality and data scarcity, variability, and noise. These attributes may cause class prediction models to be unstable and lack adequate statistical power. To address these challenges methods for integrative analysis of gene expression datasets have since been proposed.

2.3 INTEGRATIVE ANALYSIS OF GENE EXPRESSION DATA

Considering some of the challenges inherent in class prediction analysis that were highlighted in the previous section, integrative analysis of multiple studies, which ask the same/similar biological question holds promise towards high predictive performance, consistency, and robust classification models. Integrative analysis of microarray data can be viewed from two angles: *integrative data analysis* and *integrative modeling*. The former fuses two or more expression matrices into one big dataset, while the latter combines information from different expression matrices into one model. In addition, for most practices, the former is a required input to the latter. The next sections highlight major frameworks that have been developed for both integrative data and modeling analyses.

2.3.1 Integrative data analysis

Integrative data analysis of multiple and independent microarray studies can be carried out with two main approaches: *meta-analysis* and *analysis by data merging* [85]. With the meta-analysis approach, integration occurs at the interpretive level, where results (e.g., classification accuracy, p-values, ranks, etc.) from individual studies are combined. By contrast, the merging approach combines two or more microarray data by rescaling their expression values into numerically comparable measures.

2.3.1.1 Meta-analysis involves the quantitative review and synthesis of different, but related microarray studies. With a plethora of gene expression datasets available in public repositories, meta-analysis has emerged as the most popular technique to compare microarray studies at the interpretation level [21]. Ramasamay *et al.*'s [20] proposed seven-step guideline has been adopted as the *de facto* standard for conducting microarray meta-analysis.

According to a comprehensive review by Tseng *et al.* [21], several methods for microarray meta-analysis have been proposed, developed, and applied. Majority of these methods can be grouped into three main groups according to the type of statistics they combine namely, p-values (e.g., Adaptively weighted (AW) Fisher [86]), effective sizes (e.g., Random effects model [87]), and ranks (e.g., RankProd and RankSum [88]). In terms of hypothesis tests, they can be further classified into two complementary groups. In the first, the goal is to determine results, say differently expressed genes, which have a nonzero effective size in all studies [21]. Considering K microarray studies, the hypothesis can be stated as:

$$H_0 : \cap_{k=1}^{K}\{\theta_k = 0\} \text{ versus } H_a : \cap_{k=1}^{K}\{\theta_k \neq 0\} \tag{2.1}$$

where θ_k denotes the effective size of the kth study. By contrast, the hypothesis setting of the second group seeks to determine differentially expressed genes that have a nonzero effective size in at least one study. Similarly, the hypothesis can be stated as:

$$H_0 : \cap_{k=1}^{K}\{\theta_k = 0\} \text{ versus } H_a : \cup_{k=1}^{K}\{\theta_k \neq 0\} \tag{2.2}$$

Equation (2.1) is a more appropriate approach for integrative studies whose aim is to identify candidate biomarkers that are consistent and conserved, while eq. (2.2) is suited for

the identification of study specific biomarkers, particularly for studies where relatively high degrees of heterogeneity are suspected. Because eq. (2.1) can be very conservative when a lot of studies are integrated, Song and Tseng [89] relaxed the constraint of nonzero effective size in "all" studies to "majority" of studies. The modified hypothesis can be viewed as:

$$H_0 : \cap_{k=1}^{K}\{\theta_k = 0\} \text{ versus } H_a : \sum_{k=1}^{K} I\{\theta_k \neq 0\} \geq r \tag{2.3}$$

where r is a user specified parameter to indicate "majority". For example, $r \geq 0.6K$, means identify differently expressed genes with an effective size in at least 60% studies.

In spite of its popularity, inherent intricacies of microarray studies, like noise, biological and experimental variability, variability in platforms and experiment protocols, may cause meta-analysis studies to yield false positives and discordant results [22, 85]. However, most of these variability issues can be ameliorated by data merging methods.

2.3.1.2 Data merging methods, contrary to meta-analysis techniques, integrate gene expression data of independent, but related, studies into a large data matrix after transforming the expression values of the individual studies into numerically comparable values. To achieve this, specific data transformation techniques, like normalization, are applied to the gene expression data. The merged data matrix then becomes the input to further data analysis such as class prediction tasks.

Several techniques for transforming multiple gene expression data have been developed and applied in literature [22]. The most common of such methods is normalization. Hwang *et al.* [90] used normalization to combine expression values of each gene across samples from *cDNA* and *Affymetrix* platforms such that the mean and standard deviation of each gene is equal to zero and unity, respectively. Similarly, Cheadle *et al.* [91] had earlier applied the *Z-score transformation* method to transform the intensity values of *cDNA* microarray. Meanwhile, methods that are more sophisticated have since been proposed.

Using the distance weighted discriminant (DWD) method [92]—an adaption of SVM—Benito and Co. [93] integrated *Agilent* oligonucleotide data with *cDNA* data by ameliorating any potential systematic biases lurking in the datasets. Furthermore, Johnson and colleagues [94] applied an *Empirical Bayes method* (COMBAT) to transform gene expression data from

independent but related studies to have equal (similar) mean and/or variance for each gene. While this method enables data comparison, it does not eliminate any biological signal of interest nor affect the data distributions across the different studies.

Jiang *et al.* [95] proposed a distributions transformer (*disTran*) which can be used to transform two *Affymetrix* chip types such that the empirical distributions of two lung cancer datasets could be identical, so that they can be integrated. They reported that *disTran* could provide improved consistency of expression profiles across multiple datasets. *Cross-platform normalization (XPN)* [96] is also another method that combines two or more gene expression datasets into a single expression matrix. Based on a block linear model, *XPN* identifies homogeneous clusters of genes and samples across studies that have similar expression profiles. What is more, *Normalized Linear Transform (NLT)* [97] allows samples from two different microarray platforms to be linearly mapped such that the numerical range of a gene's expression values can be identical across the platforms. The mapped data can be further combined and transformed via standard normalization or *Z-transform*.

In conclusion, many more techniques exist for gene expression data integration (e.g., see [22,59,98] for and empirical comparison of various methods), and they provide an essential step in further analysis of gene expression data such as integrative modeling.

2.3.2 Ensemble & Integrative modeling

The process of seeking advice from multiple experts in order to make a better-informed decision is second nature to humans. Several methods in ensemble learning and integrative modeling have been proposed to automate such processes. The overarching assumption is that: using information (e.g., prediction) from multiple models is superior to that of a single model. In general, most ensemble learning methods [99–102] combine multiple models that are learned from a single data source. On the other hand, the multi-view learning approach (e.g., data fusion [103]) builds models by combining information from multiple data sources.

Combining information from independent, but related studies, have been proposed as a viable strategy to improve diagnostic and prognostic performance in several biomedical studies, especially cancer [104–106]. For instance, molecular, clinical, and histopathological

information can be combined to predict tumor progression in lung cancer [107].

Generally, the integrative approach to a class prediction task can be designed with two main intents: *subjective* and an *objective* bent. The subjective approach is driven by priors, which are biased by strong user and/or domain assumptions about biomarkers and their relationships with outcomes. For example, specific clinical features, risk factors, and/or molecular markers can be considered as inputs into predictive models because they have been gleaned from a domain expert and/or literature as relevant for a disease diagnosis or prognosis. By contrast, the latter framework relies on computational techniques to automatically identify and combine relevant information from different information sources (e.g., 'omic' datasets, biomedical knowledge bases, or even related models) to build a predictive model for disease diagnosis or prognosis.

Combining information from multiple sources for predictive analysis has several advantages. Since most 'omic' data are characterized by small sample size and large variable space, classification models built on them might be noisy. In addition, using different techniques to build classification models on the same dataset can yield non-unique error rates. However, combining information from the different models can complement each other so that the overall predictive error rate can be reduced potentially [83].

Several strategies for integrative model design have been proposed for integrative 'omic' data analysis. Azuaje [104] categorized them into five main groups. The strategy of the first group is to aggregate variables from different datasets, through set union or intersection, before a prediction model is learned. Naively merging datasets based on common variables can hamper predictive performance, so more sophisticated methods in *meta-analysis* and *cross-platform merging* have since been proposed [21,85]. For example, Warnat *et al.* [108] applied a cross-platform analysis (*Quantile Discretization* and *Median Rank Scores*) to combine multiple cancer microarrays into an input matrix for an SVM classifier. In the second category, different classification models are learned on homogeneous datasets (i.e., multiple gene expression datasets containing the set of genes), followed by combining the resulting models into a generalized model. Zhang *et al.* [109] used this strategy to improve the performance of a prognostic model for breast cancer patients. The third category applies feature-engineering methods to integrate heterogeneous datasets (e.g., gene expression, CNV, DNA methylation,

etc) into a unit matrix before building the model. For example, Daemen *et al.* [110] applied a kernel method [111] to transform diverse 'omic' datasets (i.e., gene expression and CNV) into a 'kernel matrix' before building a prognostic SVM model for prostate and rectal cancers. In the fourth category of strategies, predictive models are learned on heterogeneous datasets and/or information sources in a parallel fashion, followed by fusing the resulting models into a global one. This category can involve the integration of 'omic' datasets that are annotated by interrelated information sources (e.g., pathways). For instance, Ptitsyn *et al.* [112] combined microarray datasets and information from pathway databases to predict metastatic progression in cancer. The last category involves the combination of multiple datasets and/or predictive models in a serial fashion [113].

Majority of the methods that have been described above generate models that are difficult to interpret. This work contributes alternate approaches for tackling the integrative modeling tasks of related gene expression datasets—with an emphasis on model interpretability. Nonetheless, it employs a combination of some of the integrative modeling strategies that have been illustrated above. For instance, it employs both the serial and parallel paradigms at some stages of the frameworks. At one point, it builds predictive rule models on source datasets in parallel fashion, while at another point it combines information from the resulting source models to build a new target model in a multi-step, serial integration fashion. The mechanism of integration is based on the concept of transfer learning.

2.4 TRANSFER LEARNING

Transfer learning is the ability of a system to recognize and adapt knowledge/skills learned from previous tasks to a novel one. Humans, in general, will more likely solve new problems in a much faster time and with better solutions if they apply knowledge learned from previous but related tasks. For instance, skills learned from driving a sedan could be transferred easily to learn how to drive a truck. Alternatively, a farmer could leverage skills learned from growing apples and oranges to grow mangoes in a much faster time and may gain a higher yield at harvest time. This inherent feature of humans has inspired the concept of

transfer learning in the machine learning community. Several machine-learning methods, which are used for microarray class prediction task, learn new models from scratch without regard for knowledge gained previously.

Transfer learning tries to tackle the challenge of how to leverage knowledge gained from related source domains to maximize accuracy and efficiency of learning in a new target domain [24,114]. It is an effective technique for boosting learning efficiency and performance in situations where training data are scare, but other related data are available. This characteristic data requirement particularly suits gene expression as discussed previously. Meanwhile, majority of the integrative analysis of gene expression data discussed above cannot explicitly transfer vital information from multiple tasks to boost the analysis of another. Before we discuss how the transfer learning concept can be applied for integrative modeling of two or more gene expression datasets, let us first define a transfer learning task.

2.4.1 Defining the transfer learning task

Based on Pan and Yang's definition [24], let us first consider the following notations. Assume that a learning domain (e.g., microarray study for breast cancer), D, comprise a feature space,χ, and a marginal probability distribution $P(X)$, where $X = \{x_1, ..., x_n\} \in \chi$. Here, χ is the space for all genes captured by the microarray, x_i denotes the i-th gene vector corresponding to some sample i, and X symbolizes the expression matrix for the learning sample. Two domains (e.g., breast cancer and lung cancer) can be said to be different if they have different feature spaces or different marginal probability distribution [24].

Consider a given domain, $D = \{\chi, P(X)\}$, and a class prediction task, $T = \{Y, f(.)\}$, where Y denotes the label space (e.g., "Primary Tumor" or "Normal" for a binary classification task), and $f(.)$, represents the objective function or rules that is yet to be determined. In addition, let us consider a *source domain dataset*, $D_T = \{(x_{S_1}, y_{S_1}), ..., (x_{S_{n_S}}, y_{S_{n_S}})\}$, where $x_{S_i} \in \chi_S$ and $y_{S_i} \in Y_S$ represent a data instance and a corresponding class label, respectively. Similarly, a *target domain dataset*, can be defined as $D_T = \{(x_{T_1}, y_{T_1}), ..., (x_{T_{n_T}}, y_{T_{n_T}})\}$, where $x_{T_i} \in \chi_T$ and $y_{T_i} \in Y_T$, respectively, represent data instance and the corresponding class label for the target data.

Using the above notations knowledge transfer from a single source domain to a single target domain can be defined as follows [24]:

Definition 2.4.1 (Single source transfer)**.** *Consider a source domain D_S and a source learning task, T_S, a target domain, D_T, and a target learning task T_T, the aim of a single-source knowledge transfer is to improve learning of the target predictive function $f_T(.)$ in D_T, while incorporating knowledge gleaned from D_S and T_S, where $D_S \neq D_T$ or $T_S \neq T_T$.*

Similarly, we can define knowledge transfer from multiple source domains to a single target as follows:

Definition 2.4.2 (Multiple source transfer)**.** *Consider multiple source domains $\{D_{S_1}, ..., D_{S_N}\}$ with corresponding source learning tasks, $\{T_{S_1}, ..., T_{S_N}\}$, a target domain, D_T, and a target learning task T_T, the aim of a multi-source knowledge transfer is to improve learning of the target predictive function $f_T(.)$ in D_T, while incorporating knowledge gleaned from $\{D_{S_1}, ..., D_{S_N}\}$ and $\{T_{S_1}, ..., T_{S_N}\}$, where $\forall k \in \{1, ..., N\}, D_{S_k} \neq D_T$ or $T_{S_k} \neq T_T$.*

A complex scenario of knowledge transfer can involve multiple source domains and multiple target domains; however, for brevity, the above definitions will suffice for this discussion.

From the definitions above, note that the condition $D_S \neq D_T$ implies that either $\chi_S \neq \chi_T$ or $P(X_S) \neq P(X_T)$ are different (e.g., different array platforms, different disease type, different study cohort). Similarly, the condition $T_S \neq T_T$ implies that either $Y_S \neq Y_T$ or $P(Y_S|X_S) \neq P(Y_T|X_T)$. That is, either the label spaces between the two domains are different (e.g., source domain has binary class like *primary tumor* vs *normal tissue*, while target domain has 4 classes like stages of cancer), or the user defined classes are unbalanced in the two domains.

Finally, when there exist some form of relationships—implicit or explicit—between the variables spaces of the source and target domains, then they are said to be *related*. Identifying and abstracting this relatedness is key to effective knowledge transfer, and the basis for integrative modeling via knowledge transfer. For instance, majority of transfer learning algorithms leverage the identified relatedness in order to decide on "what" information to transfer and "how" to transfer them. Meanwhile, the question of "when" to transfer is an open research problem.

The goal of transfer learning is to improve the learning performance on the target task. **Positive transfer** occurs when the transferred knowledge from the source(s) improves learning performance on the target, while **negative transfer** is the reduction of learning performance on the target after knowledge transfer. Rosenstein *et al.* [115] showed that there is a correlation between relatedness and negative transfer. That is, if two domain and/or tasks are too dissimilar negative transfer may occur. Given two or more source datasets, it will be desirable to determine their relative relatedness to the target dataset, and how the relatedness can be used to avoid negative transfer (i.e., *bad advice*). The proposed dissertation project will explore mechanisms for estimating relative relatedness between the source(s) and target; and how it affects positive/negative transfer.

2.4.2 Categorization of transfer learning

Based on different conditions between the source and target domains and tasks, Pan and Yang [24] categorized transfer learning settings as inductive transfer learning, transductive transfer learning, and unsupervised transfer learning (see table 1).

In the case of inductive transfer learning, the target task is different from the source task, regardless of whether the source and target domains are the same or not. For instance, in the case where the target task is a classification or regression one, inductive transfer learning here aims at achieving a high classification performance by transferring knowledge from the source task, say another classification task.

For transductive transfer learning, the source and target domains are different while the source and target tasks are the same. In this setting for instance, knowledge could be transferred between domains with different but related feature spaces and marginal probability distributions of input data between the source and target domain. This kind of transfer learning is referred to as domain adaptation [116].

Unsupervised transfer learning is similar to an inductive transfer learning setting (i.e., the target task is different but related to the source task), but the focus is to solve unsupervised learning tasks in the target domain, for example clustering and dimensionality reduction [117, 118].

Table 1: Different Setting of Transfer Learning [24]

Transfer Learning Settings	Related Areas	Source Domain Labels	Target Domain Labels	Tasks
Inductive Transfer	Multi-task Learning	Available	Available	Regression, Classification
	Self-taught Learning	Unavailable	Available	Regression, Classification
Transductive Transfer	Domain Adaptation, Sample Selection Bias, Co-variate Shift	Available	Unavailable	Regression, Classification
Unsupervised Transfer		Unavailable	Unavailable	Clustering, Dimensionality Reduction

Based on these different transfer-learning settings the notion of what knowledge to transfer between related domains has inspired different transfer learning approaches. The question on "what to transfer?" between related domains has driven the development of algorithms and transfer learning research for many years [119]. Based on what information to transfer, transfer-learning approaches can be summarized into instance-based transfer, feature representation-transfer, parameter-transfer, and relational-knowledge transfer (see table 2).

In cases where the instance space between the source and target domain are the same or related, certain transfer learning approaches, also known as instance transfer, assume that some parts of the source domain data could be reused for learning in the target domain. Some techniques that have been used in this approach are instance re-weighting and importance sampling [120, 121].

For cases where the feature space between the source and target domains are related, some transfer learning approaches aim to learn good discriminatory feature representation for the

Table 2: Different Approaches to Transfer Learning [24]

Transfer Learning Approach	Description
Instance-based transfer	Re-weighing of labeled data in source domain for use in target domain (e.g., [120, 121])
Feature-representation-based transfer	Identify specific features ("pivot") that tends to reduce the difference between source and target domains (e.g., [122, 123])
Parameter-based transfer	Identify priors or parameters that the source and target domain models share in common (e.g., [124, 125])
Relational-knowledge transfer	Relational knowledge between the source and target domains are captured to facilitate transfer (e.g., [126, 127])

target domain. In other approaches, discriminatory features are transferred across domains through a mapping function learned between the feature space of the source and target domain. This type of transfer can be referred to as feature-base or feature-representation transfer approach [122, 123, 128].

Driven by parametric statistical models, certain transfer learning methods assume that the source and target learning tasks share some common parameters or prior distributions of hyper-parameters of the models. In this approach, knowledge could be transferred across domains by discovering the shared priors or parameters. This case is also referred to as the parameter-transfer learning approach [124, 125].

Finally, other transfer learning approaches deal with knowledge transfer between relational domains. The assumption is that, in relational domains the relationships among objects within the source and target domains are similar. Some transfer learning approaches learn the structures of these relationships before transfer. This case of transfer learning, also referred to as relational knowledge transfer, is predominantly driven by Statistical Relational Learning techniques [126, 127].

2.4.3 Transfer Rule Learning

Transfer learning techniques have been applied to a slew of real-world applications, including knowledge discovery problems in biomedical research. See [24, 129, 130] for a comprehensive list of examples, as well as the benefits of transfer learning. Majority of transfer learning applications are based on algorithms (e.g., artificial neural networks (ANN) or SVMs), that generate models which are difficult to interpret by humans, use relatively large number of variables, or are computationally intensive to train, and might therefore not be suitable for 'transcriptomic' data modeling.

Ganchev [28] proposed a novel framework for transfer learning, called TRL, which is particularly well suited for integrative biomarker discovery from related but separate biomarker profiling studies. Based on the transfer-learning concept, TRL learns modular and interpretable rules from the source data, and uses them to aid learning of a new classification rule model on the target data. TRL proposed two methods of knowledge transfer namely, *whole-rule* and *structure* transfer. The former employs a strict incorporation of source variable values in the transferred rules, while the latter, which transfers rules without variables, injects flexibility into rule induction on the target dataset. Due to the numerous forms of variability inherent in different 'transcriptomic' studies, such as gene expression, whole-rule structure should be used with caution.

The proposed multi-source transfer rule-learning frameworks vastly extends the foundation set by TRL. Unlike TRL, it learns and transfers prior rules from multiple datasets. This is akin to seeking 'advice' from multiple experts. They can track the relative contribution of knowledge from each source, which may give an indication of any potential negative transfers. While TRL relies on identical variables between source and target domains for knowledge transfer, the frameworks proposed herein leverages a more biologically intuitive mechanism, functional modules, to capture and abstract the relatedness between the source(s) and target domains. In addition, the proposed framework is flexible such that, theoretically, it can be used for integrative modeling spanning disparate domains. Finally, the proposed framework, like TRL, is based on the classification rule learner (RL) [18].

2.5 FOUNDATIONS OF RULE LEARNING WITH RL

RL, a descendant of the Meta-DENDRAL family of inductive rule learning systems [131,132], was used as the building blocks for this work due to several properties that make it particularly suitable for predictive modeling of gene expression datasets [28]. First, unlike other knowledge discovery algorithms like ANN or SVMs, humans can easily interpret classification models learned by RL. Second, RL is simple and flexible such that, users can leverage domain knowledge to set learning parameters *a priori* in order to improve a search in the hypothesis space. Third, RL covers rule with replacement. That is, it does not recursively partition the instance space of the training example (e.g., C4.5 [17]), nor does it eliminate training instances covered by a rule as learning proceeds (e.g., CN2 [133]), but instead it allows rules to cover overlapping regions in the instance space. Covering training instances with replacement particularly suits situations where data are scarce (e.g., microarray data), since ample data would be available to provide statistical support for newly induced rules. Fourth, RL can handle nonlinear relationships as well as hierarchical variables, such as cancer and its subtypes. Fifth, to avoid costly errors, RL can abstain (i.e., it is agnostic) from predicting a test case when it has low confidence in the accuracy of the rule [29]. It has therefore been used successfully in several classification tasks involving genomic and proteomic studies [29, 31, 32, 134, 135].

The following sections provide an overview of the core concepts, which underpins inductive rule learning with RL. They describe the general formulation of a rule learning problem; data and hypothesis representation; the process of learning classification rules; and making inference with the learned rules.

2.5.1 Problem Formulation

A classification rule learning task, on a given gene expression data set (see section 2.2), can be formally summarized as shown in fig. 2. Given the data, find a set of classification rules that can accurately classify new instances. The input involves representation formalisms for describing, respectively, the gene expression data (i.e., expression matrix) and the induced

Input:

▷ a data description language; i.e., defining the form of data set

▷ a hypothesis description language; i.e., defining the form of rules

▷ a coverage function, *COVERED(r, e)*, defining whether rule r covers example e

▷ a class variable, C

▷ a set of training examples, $\mathcal{X}$, described in the data description language

Output:

▷ a set of rules, $\mathcal{R}$, formulated in the hypothesis description language such that:

▷▷ it is *complete*, i.e., covers all examples in $\mathcal{X}$, and

▷▷ it is *consistent*, i.e., correctly predicts all examples in $\mathcal{X}$

Figure 2: Formulation of a rule learning task; adapted from [19]

set of rules (i.e., rule model). To connect a rule model with the data description a *coverage function* is desired. A rule is said to *cover* a data instance if it logically satisfy the description of the instance. It should be noted that the notions of *completeness* and *consistency* as defined in fig. 2 are idealistic. Normally, inductive rule learners, like RL, employ heuristics to search the space of rules to optimize these terms.

2.5.2 Data Representation

Gene expression datasets are comprised of hundreds or thousands of measured variables, most of which are irrelevant [136]. Majority of machine learning algorithms were not originally designed to cope with these large amounts of irrelevant variables, which may degrade model performance [137]. Due to the curse of dimensionality, predictive models can over-fit gene expression data. Meanwhile, one of the most important goals of predictive modeling is to identify and select a handful of relevant variables from among the thousands that can accurately predict a disease state or estimate the risk of disease in an individual. The selected variables serve as building blocks for constructing classification models. Therefore,

variable selection is a crucially important component of predictive rule modeling for gene expression datasets. What is more, variable selection can also facilitate data visualization and data understanding, provide faster and more cost effective models, enhances results comprehensibility, and improve model performance.

A variety of variable selection techniques have been proposed for 'transcriptomic' data. Majority of these methods can be classified into three main groups namely *filter methods*, *wrapper methods*, and *embedded methods*. Filter methods employ a variety of variable relevance score (e.g., correlation), which is based on intrinsic properties of the data, to remove irrelevant variables. The wrapper methods incorporate a classification model to search the space of subsets of variables to find variables that maximizes performance of the model. For the embedded methods, the search for an optimal subset of variables is incorporated into model development. See [137] for an in-depth review on variable selection methods for biomedical datasets.

An example (i.e., observed) gene expression data consists of $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, where each x_i is a vector, length p, of expression values, and y_i is a scalar or qualitative value denoting the target class value. X is continuous, which a large number of symbolic data mining algorithms, like RL, are unable to handle and require discrete data [134, 138, 139]. Therefore, the data must be transformed and represented in a discretized form.

Discretization, the process of converting a continuous variable to a discrete one, has several advantages. It widens the space of knowledge discovery algorithms that can be applied for modeling. By transforming numerical variables into nominal ones, it also serves as a data reduction method that can aid data visualization and interpretability. It has been shown to improve the classification performance of most algorithms, including SVM or Random Forest, which can handle continuous data [140]. In addition, since 'transcriptomic' data can be characterized by noisy and redundant variables, discretization can be used as a variable selector to weed out irrelevant variables. For instance, variables that are discretized into a single interval can be filtered out as irrelevant since they cannot discriminate the target class variable. One notable disadvantage about discretization is that, theoretically, it can lead to information loss, which can also reduce classification performance [141].

Several techniques have been proposed for discretization [139]. These techniques can

be classified as supervised or unsupervised. Supervised methods (e.g., Fayyad and Irani [142] and EBD [141]) use information about the target class variable for discretization, while the unsupervised techniques (e.g., Equal-width and Equal-Frequency [143]) do not. In addition, discretization techniques can also be categorized as univariate or multivariate. Univariate techniques (e.g., EBD) discretize continuous-valued variable independently of all other predictor variables, while the multivariate methods (e.g., [144]) consider the interaction among predictor variables during discretization. See [139] for an in-depth review on the taxonomy and empirical analysis of discretization techniques in supervised machine learning.

2.5.3 Rule Representation

Given a set of training examples, RL learns a disjunctive set of conjunctive **IF-THEN** rules, each of which has the form:

IF *Antecedent* THEN *Consequent*

where the *Antecedent* consists of one or more variable tests, which can be called conjuncts, and the *Consequent* denotes prediction of the target class variable. Every induced rule has classification-relevant statistics associated with it. Let us, for instance, consider the hypothetical rule below:

`IF ((gene1 > 1680) AND (gene2 ≤ 28.6)) THEN (Class = Case)`
`CF = 0.98, PV = 0.007, TP = 56, FP = 4`

where `gene1` and `gene2` are biomarkers with two intervals of values. This rule can be interpreted as follows: "when `gene1` is up-regulated (i.e., value $>$ 1680) and `gene2` is down-regulated (i.e., value $\leq$ 28.6), then predict the target class as Case." In addition, relevant statistics are associated to each rule induced by RL. In the given example above, the ensuing statistics mean that RL induced the rule with a 98% degree of confidence, which we call the **Certainty Factor (CF)**. Frequently used measures to estimate CF are precision, information gain, Laplace estimate, etc [19]. **PV** represents the p-value, which can be estimated by *Fishers exact test* or *likelihood ratio statistic* [133]. **True Positives (TP)** are the number of examples that are correctly covered by a rule (i.e., it covers the example, and its consequent

equals the class label of the example). **False Positives (FP)**, on the other hand are the number of examples that are incorrectly covered by the rule (i.e., it covers the example, but its consequent does not equal the class label of the example). $TP + FP$, the fraction of training examples covered by the rule, is also known as **Coverage**.

2.5.4 Learning a Rule Model with RL

Figure 3 depicts a pseudocode for the heuristic rule-space search employed by RL. Internally, RL stores induced rules in a priority queue (aka, the beam) by sorting them according to their CF and coverage. Given a set of training examples and user specified constraints, the algorithm proceeds as a heuristic beam search through the space of rules, using a general-to-specific approach [145].

First, it considers every variable as a potential predictor of the target class variable. For each variable value, it creates as many rules as there are target class values. Example, for a Case/Control binary class, it will create two rules for each variable value. One rule predicts Case and the other predicts Control. Second, it places an induced rule on the beam if it is *interesting* and satisfies user-specified constraints, also known as good-rule criteria. The constraints are **minimum CF**, **minimum coverage**, **maximum FP**, **beam width** (i.e., maximum number of rules allowed on the beam), **inductive strengthening** (i.e., the number of previously uncovered instances that a newly induced rule must cover), and **maximum conjuncts** (i.e., the maximum number of *variable-value* pairs allowed in a rule antecedent.

Subsequently, each rule on the beam is specialized if it satisfies the constraints. Specialization is the process of adding conjuncts to the rule antecedent until the constraints are violated. The algorithm stops and outputs the set of rules on the beam if there are no more rules to specialize. This set of classification rules output by RL is referred to as a rule model, which can be used to classify an unobserved data instance.

```
1: function BEAM–SEARCH(D, C)
2:     ▷ D : a set of training examples
3:     ▷ C : user specified constraints for rule learning
4:     interesting_patterns ← ∅
5:     new_beam ← {∅ ⇒ class_1, ∅ ⇒ class_2, ...}
6:     beam ← ∅
7:     while new_beam ≠ ∅ do
8:         beam ← new_beam
9:         new_beam ← ∅
10:        for all rule ∈ beam do
11:            S ← SPECIALIZE(rule)
12:            for all s ∈ S do
13:                if ISRULEINTERESTING(s, IC, D) then
14:                    ▷ IC : user specified interestingness criteria
15:                    interesting_patterns ← interesting_patterns ∪ s
16:                end if
17:                if ISGOODRULE(s, C, D) then
18:                    new_beam ← new_beam ∪ s
19:                end if
20:            end for
21:        end for
22:    end while
23:    return beam
24: end function
```

Figure 3: Pseudocode for a heuristic rule-space search with RL

2.5.5 Classification & Conflict Resolution

In using the rule model to predict the class of a new instance, two problems may occur. First, none of the rules may *fire*, that is, cover the instance. Second, RL learns rules with replacement, which means that the rules are not mutually exclusive with respect to the instances. Therefore, multiple rules could cover the same instance, resulting in potential conflicting classification.

Several strategies can be adopted to address these problems. With the first problem, a default rule, which predicts the majority class, can be added to the model. Typical solutions for the second problem involve voting mechanisms and individual rule statistics. Below are some strategies used for resolving such conflicts [146]:

1. **First matching rule**

   Since the rules are sorted in the order of **CF** and coverage, this strategy selects the rule with the highest confidence.

2. **Equal voting**

   Here, every matching rule contributes a single vote for its class; the class with majority wins.

3. **Weighted voting**

   For this strategy, each matching rule votes with a weight of its confidence; the class with the highest summed **CF** and/or **Coverage** wins.

4. **Lowest FP**

   The class of the rule with the lowest **FP** wins.

5. **Minimum P-value**

   The class for the rule with the least p-value (i.e, most relevant) is selected.

3.0 METHODS

The proposed methods are types of incremental learning. Unlike the traditional approach where learning from new data normally involves discarding all existing classifier inputs by retraining a new one—a phenomenon known as catastrophic forgetting [15, 147]—TRL-FM, KARL, MS-TRL, and iTRL are aimed at leveraging previously acquired information from multiple sources, i.e., classifiers and biological knowledge sources, to augment learning of a new classifier. Though conceptually simple, these algorithms provide significant extensions to TRL and its predecessor, RL. Section 3.1 recounts a published work (TRL-FM), which results motivated the creation of subsequent methods. Section 3.2 presents KARL, which pools background knowledge from multiple sources like literature, knowledge bases, or domain experts to augment predictive rule modeling. In contrast, the notion of multiple sources, as employed by MS-TRL, involves multiple rule models developed from related gene expression studies; Section 3.3 describes MS-TRL. Finally, section 3.4 concludes this chapter with a generalized framework for multi-source rule modeling via incremental transfer learning—iTRL.

3.1 TRL-FM

3.1.1 Background

Knowledge transfer can be facilitated in several ways. For instance, you can leverage the relatedness between the source and target to facilitate knowledge transfer. TRL's mechanism of capturing relatedness is to identify common variables between the source and target

datasets. Although this mechanism was able to facilitate knowledge transfer, however, it could be improved. Studies have shown that different classification models built on independent microarray datasets can contain different sets of biomarkers with little overlap. In addition, models based on different variable sets can yield similar classification performance when tested on the same validation dataset [25]. This means that relying solely on identical variables to establish commonality might not be enough, and therefore exploring and incorporating other means of determining variable equivalence could be vital for model performance. To that end, we developed a methodology that leverages transfer rule learning and functional modules (FMs)—two or more genes that are related to the same/similar biological process—that we call TRL-FM [135], to capture and abstract domain knowledge in the form of classification rules to facilitate integrative modeling of multiple gene expression data.

Our goal in this study was threefold. First, to test whether FMs can be used to capture the underlying commonality among variables of different but related gene expression datasets, and are more effective when used as bridges to assist knowledge transfer than relying on identical variables. Second, to test the hypothesis that "integrative modeling via the TRL-FM approach outperforms traditional models based on single gene expression data sources." Last, to evaluate and compare the classification performance of TRL-FM with traditional methods, using 21 gene expression datasets that were collected from three respective studies: one on brain cancer, one on prostate cancer, and one on a lung disease (idiopathic pulmonary fibrosis or IPF).

3.1.2 TRL-FM: The framework

Figure 4 depicts an overview of the TRL-FM framework. For the sake of simplicity, this framework performs transfer between two different, but related, sets of microarray data, i.e., a source and a target. The key steps according to the framework are as follows: First, select relevant variables from the source(s) using a feature selection method such as EBD, i.e., feature selection via discretization. Second, identify FMs among the selected variables. Third, using the discovered FMs, along with rules induced from the source(s) datasets; build

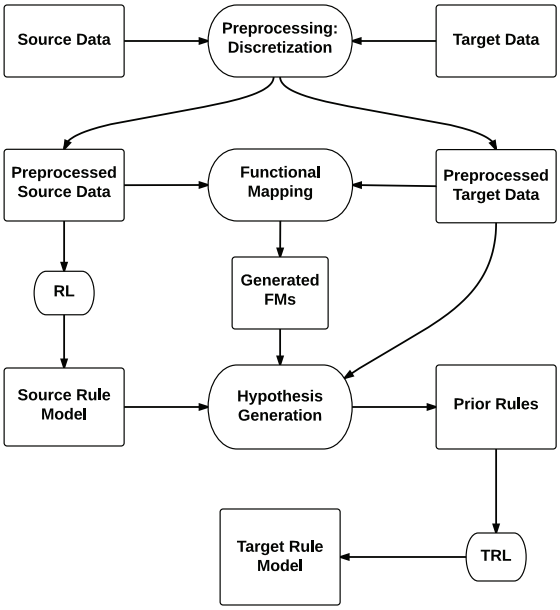

Figure 4: An illustration of a framework for transfer rule learning via functional mapping, TRL-FM

a prior hypothesis of classification rules. Finally, using the prior hypothesis as a seed, learn a new classification rule model from the target dataset.

Figure 5 illustrates a pseudocode for implementing the TRL-FM framework. Major components of the algorithm are, (1) a sub-routine to map and abstract relatedness among a set of given variables, (2) a prior-rules generation engine, and (3) a modified heuristic beam search with prior rules (TRL). Given a source dataset (D_s), a target (D_t), and a set of user specified constraints for rule learning (C), as inputs, the algorithm proceeds as follows.

First, the feature selector (i.e., discretization with **EBD** in this case) selects relevant variables from both the source(s) and target datasets (see line 4, fig. 5) for functional mapping and abstraction of variable relatedness (see line 5, fig. 5). Note that given a list of arbitrary domain variables, several methods can be used to define and abstract their relatedness. For instance, with a list of genes, relatedness could be defined as association to a common

1: **function** TRL–FM(D_s, D_t, C)
2: ▷ D_s : source dataset; D_t : target dataset
3: ▷ C : user specified constraints for rule learning
4: $S_v \leftarrow \text{EBD}(D_s)$; $T_v \leftarrow \text{EBD}(D_t)$
5: $FM_s \leftarrow$ MAPFUNCTIONALASSOCIATIONS(S_v, T_v)
6: $S_M \leftarrow \text{RL}(D_s, C)$
7: $S_{M_V} \leftarrow$ GETMODELVARIABLES(S_M)
8: $prior\_rules \leftarrow$ GENERATEPRIORRULES(S_{M_V}, D_t, FM)
9: $model \leftarrow \text{TRL}(prior\_rules, D_t, C)$
10: **return** *model*
11: **end function**

Figure 5: A pseudocode for implementing the TRL-FM framework

molecular function, pathway, or disease. TRL-FM in particular uses a Gene Ontology (GO)-similarity-based method to identify and abstract relatedness among genes (see section 3.1.3). Given a list of genes, this method outputs FMs, clusters of functionally related variables that facilitates the generation of prior rules (see line 8, fig. 5, and section 3.1.4 for details). Finally, with the prior rules as seed, target dataset, and user the user specified criteria for good rules, the algorithm induces a new rule model using the TRL sub-routine.

3.1.3 Identifying GO-Based Functional Modules

The major contribution of TRL-FM was the application of GO-based FMs to facilitate the identification of functionally related variables for transfer rule learning. The Gene Ontology is a representation of specific domain knowledge in cell biology. It is represented by a directed acyclic graph, where terms—description of a biological concept like a cellular process—are nodes and edges are the relationships among them [4]. It also provides an annotation knowledge base, which describes terms and the gene involved with them [148].

One major challenge with the Gene Ontology graph is that, several semantically similar terms annotate the same gene, while the same term can annotate several genes. Our goal was to avoid redundant GO annotations as well as avoiding losing sight of the many-to-many relationships between genes and terms. To that end, we clustered semantically similar terms into functional themes, and then mapped annotated genes to them.

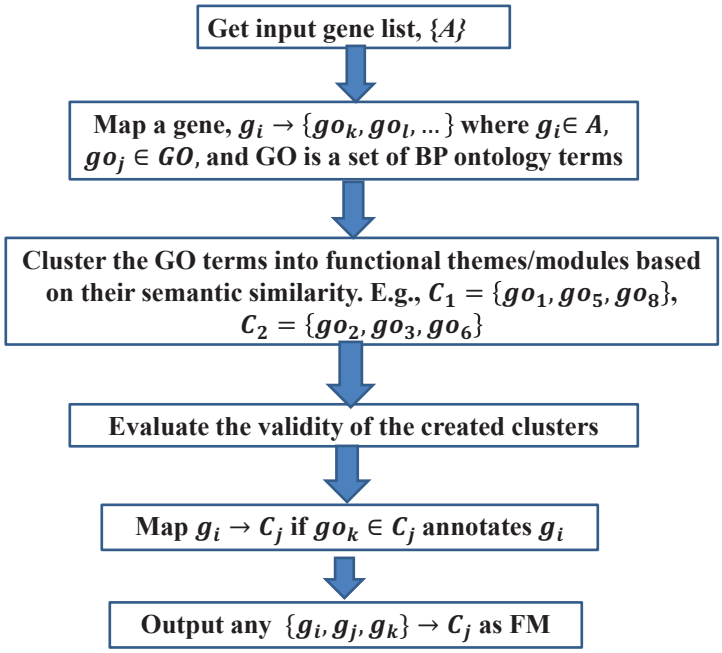

Figure 6: A protocol to identify FMs from a set of genes

Figure 6 summarizes the major steps involved in our method for capturing GO-based FMs given an input of arbitrary genes. First, we mapped each gene in the input list to the corresponding GO term(s) that annotate(s) it, according to the GO annotation database [65]. For example, if G denotes the set of input genes, then each gene, $g \in G$, is mapped to the GO term, $go \in GO$, that annotates it. Here, GO is a set which refers to terms in the biological process sub-ontology in the GO graph. For example, the mapping $M(g_1) \Rightarrow \{go_1, go_3\}$ means that terms go_1 and go_3 annotate gene g_1. Subsequently, a union of all GO terms that annotate at least one member of the input gene set is formed. This set of GO terms served as input to the clustering phase.

Second, using semantic similarity [149] as a distance measure, we constructed a similarity matrix among the GO terms. With the similarity matrix as input, we applied the spectral clustering algorithm [150] to group the GO terms into functionally similar clusters. Meanwhile, the Silhouette value technique [76] was used to estimate appropriate cluster size as well as filter out spurious clusters.

Finally, each gene, g_i, was mapped to a cluster, C_i, if there existed at least one term in the latter that annotates the former. This approach enabled us to identify groups of genes that perform the same or similar functions as well as genes that perform multiple functions. Any group of genes that mapped to a particular GO cluster (e.g., $\{g_1, g_2, g_3\} \Rightarrow C_1$) forms a functional module—the output. The FMs thus serve as bridges to facilitate the creation of prior rules for transfer

3.1.4 Prior Rules Generation via Functional Mapping

Figure 7 represents a pseudocode for the prior rule generation engine. Given an input of a set of source variables, S_{M_V}(i.e., all variables involved in a source rule model), target dataset (D_t) and a definition of variable relatedness (FM), the algorithm outputs a list of prior rules. First, using the FM as a bridge between the source(s) and target, rules are instantiated with variables of the target. For every target variable that co-exist with any source variable in an FM (see line 8, fig. 7), prior rules are instantiated. Given a variable, the INSTANTIATERULE sub-routine first formulates a rule scaffold, which is then populated with all variable and target class values. For example, assume a selected target variable, `GENE`, takes two values `Down` and `Up` , while the target class variable, `Class`, also take two values, `Case`, and `Control`. First, a rule scaffold of the form (`IF (GENE = ?) THEN (Class = ?)`) is created, and then after populated with a permutation of all variable values into the following list of rules:

```
IF (V = Down) THEN (Class = Case)
IF (V = Up)   THEN (Class = Case)
IF (V = Down) THEN (Class = Control)
IF (V = Up)   THEN (Class = Control)
```

```
1: function GeneratePriorRules(S_{M_V}, D_t, FM)
2:     ▷ S_{M_V} : a set of variables from source rule model
3:     ▷ D_t : a set of training examples from target data
4:     ▷ FM : a set of functionally related variables
5:     prior_rules ← ∅
6:     T_v ← EBD(D_t)
7:     for all t ∈ T_v do
8:         if t ∈ {S_{M_V} ∩ FM} then
9:             R ← R ∪ instantiateRule(t)
10:            for all r ∈ R do
11:                if isGoodRule(r, C, D_t) then
12:                    prior_rules ← prior_rules ∪ r
13:                end if
14:            end for
15:        end if
16:    end for
17:    return prior_rules
18: end function
```

Figure 7: A pseudocode for generating prior rules via functional mapping

This mechanism, first coined as *rule structure transfer* by Ganchev et al. [28], of instantiating prior rules has several advantages. First, it facilitates the needed flexibility for transfer learning across domains of disparate feature spaces. For domains that have even identical variables, mapping corresponding variable values can be very challenging because of unequal value distributions. After discretization, for instance, the number of intervals and the locations of cut points for identical variables can be different. Second, with this mechanism a user can formulate prior rules, culled from a domain expect or literature, and explicitly load them into the framework to seed or guide learning of new classification rules (see section 3.2).

Results from the TRL-FM work (see section 5.1) suggested that the provenance of prior rules, and how they are generated improve, significantly, the performance of transfer rule learning, particularly when there are multiple related source data available. Therefore, there was a need to investigate new mechanisms of generating prior rules from multiple sources—multiple related models, knowledge bases, or literature—to augment transfer learning of classification rules. To that end, I subsequently developed KARL, MS-TRL, and a generalized framework for incremental transfer rule learning, referred herein as iTRL.

3.2 KARL

The Knowledge Augmented Rule Learning (KARL) framework is a variant of the multi-source transfer rule learning frameworks developed herein for predictive rule modeling. There are two main differences between MS-TRL and KARL. First, the provenance of multiple information are different. In the former, prior rules are generated from information gleaned from multiple "related" rule models, while, for the latter, prior rules are generated from information that are garnered from multiple domain experts, literature, and/or multiple domain knowledge bases (e.g., Ontologies or Databases). Second, while the notion of *rule interestingness* is purely data-driven (i.e. object) in MS-TRL, KARL augments "objective" measures with "subjective" notions to induce rules.

Most knowledge discovery systems, like **RL**, can generate a slew of patterns, most of which are of no interest to the user. It is therefore important to define a measure of interestingness that could be used to filter out trivial patterns. The main methods for measuring rule/pattern interestingness is based on their properties and/or statistical strengths—objective means. In **RL**, for instance, the good-rule (interestingness) criteria is based on statistics like **Coverage**, **TP**, **FP**, and/or **Inductive Strengthening**. Other alternate means of defining interestingness are derived from a users' beliefs or expectations—subjective measures—that are specific to domain knowledge. While one user may be interested in patterns that are associated with disease causality, another may be interested in patterns that highlight drug toxicity.

3.2.1 Background

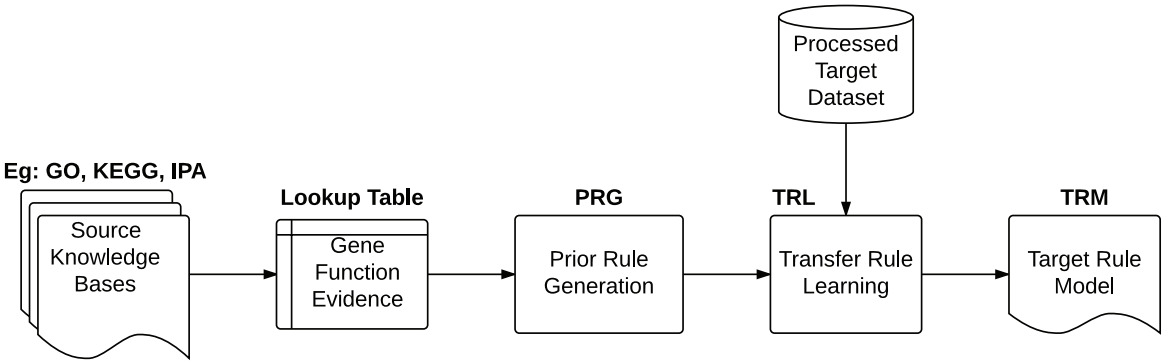

Figure 8: Knowledge Augmented Rule Learning framework

Even though objective measures are very useful—in fact, majority of knowledge discovery systems rely on them—it has been noted that they are unable to capture all the complexities within the knowledge discovery process [154,155], and therefore it is necessary to complement them with subjective measures. To this end, KARL was developed. With KARL, a pattern is interesting if there are evidence that its variables (e.g., genes) are associated with functional processes with a desirable domain. For this project, we focused on the domain, cancer, and the markers associated with its hallmarks.

Figure 8 illustrates the general framework of KARL. Its main components are, (1) the extraction of domain knowledge; (2) abstraction of the domain knowledge into a table of function evidence; (3) generation of prior rules; and (4), induction of the final rule model. Sections 3.2.2 to 3.2.5 describe these components in details.

3.2.2 Domain Knowledge Extraction

Every domain has unique characters, interactions, nuances, properties, and different degrees of complexities—some domains may even exhibit hierarchical inheritance properties. For a study of this nature, it is essential to focus on a particular domain, a sub-domain, or an aspect of a domain. This project focused on cancer, and some of its main actors—genes—that pertain to its diagnosis, prognosis, and screening.

Cancer is a leading cause of human death in the world, second behind only heart disease. According to the NCI, there are more than 100 types of cancer, and more than 500 genes involved in cancer. The explosion of cancer-related research has resulted in an exponential growth of cancer-related data from multiple resources such as scientific publications, transcriptomics, epigenomics, proteomics, GWAS, cytogenetics, etc, stored in diverse databases [156]. Due to the complexities and heterogeneity of information stored in these repositories, it is challenging to retrieve, analyze, and assimilate these data into relevant domain knowledge. Knowledge bases and resources, like GO, KEGG and IPA, just to mention a few, that employ domain experts and computational tools to integrate, curate and annotate relevant biological data from literature into plausible domain-knowledge, have alleviated these challenges. For brevity, this project adopted the Ingenuity® Knowledge Base as the resource to extract relevant domain knowledge.

3.2.2.1 Ingenuity® Knowledge Base The Ingenuity® Knowledge Base (IKB) contains evidence-based domain knowledge in the form of gene-interaction networks that were curated and verified from published literature by PhD-level scientists. The IKB is structured using an ontology, and using standard gene annotations (e.g., Entrez) each finding is categorized into three main species—human, mouse, or rat. In addition, every finding is supported by relevant literature citation(s) and a link to their respective PUBMED abstract(s).

Ingenuity® Pathway Analysis (IPA) is a tool, which is built on IKB for inference and exploratory analyses. Given a list of genes, it creates and outputs molecular networks (algorithmically generated pathways) by mapping each gene to information contained in the knowledge base. The output networks can be categorized into diseases, biological functions, or canonical pathways. Using the Hypergeometric test, it can estimate and flag significant genes based on information within the IKB. For instance, it can answer the question of what diseases, biological functions, or canonical pathways do the significant genes among the input list affect. Thus, with IPA we were able to extract domain knowledge in biological functions and networks that are associated with the hallmarks of cancer.

3.2.2.2 Hallmarks of Cancer

Though cancer is well-known to entail a lot of heterogeneity, all cancers have similar traits. For a cell to progress into a tumor, it acquires a whole gamut of aberrant properties. While different cancer types may require different combinations of these properties, typical behaviors—hallmarks—that underpin them can be categorized. Seminal work done by Hanahan and Weinberg [151, 157] has suggested that an extensive catalog of cancer cell genotypes is a manifestation of six main capabilities that turn to modify the physiology of the cell, and dictates its malignant growth.

These six capabilities' hallmarks are **sustaining proliferative signaling**, **evading growth suppressors**, **activating invasion and metastasis**, **enabling replicative immortality**, **inducing angiogenesis**, and **resisting cell death**. These hallmarks can be merged further into three main properties that enable cells to survive, disseminate, and proliferate. The first is the **faulty control of the cell cycle** (i.e., sustaining proliferative signaling and evading growth suppressors hallmarks); the second property is the **faulty control of cell death** (i.e., resisting cell death and enabling replicative immortality hallmarks); and the third property is the **invasiveness and metastatic capabilities** (i.e., inducing angiogenesis and activating invasion and metastasis hallmarks).

KARL relies on the hypothesis that, "domain characters (i.e., genes) that are associated with these properties are more likely to play a significant role in the induction of a rule model." It is this hypothesis, the basis of KARL's subjective notion of interestingness, that motivates the algorithm to augment the learning of a predictive rule model with prior domain knowledge. To generate prior rules to seed knowledge-transfer, KARL relies on IPA-based functional evidence of genes that are associated with these three main hallmarks. Thus, the algorithm requires a data structure (i.e., a lookup table that contains abstracted domain knowledge), which provides gene function evidence to augment rule induction.

3.2.3 Functional Lookup Table

The functional lookup table is a comma-separated values (CSV) file that contains vital domain knowledge, which KARL can use to instantiate prior rules or augment general rule induction. It is passed as an input parameter to the KARL algorithm. Thus the KARL

framework requires it to function, else the default rule generation engine, **RL**, is used to induce rules on the target data.

For each target dataset, a complementary functional lookup table was generated with the aid of IPA. To identify significant domain variables that are associated with domain hallmarks the whole variable set of the target dataset can be used as input to IPA. Since gene expression datasets can consist of tens of thousands of variables it is necessary to pre-filter the variable list in order to simplify the process. Both supervised and unsupervised variable selection methods, for instance, can be used for this process.

Given the input list, IPA outputs functional networks (or lists, modules) that are ranked according to a $P_{score} = -\log_{10}(p-value)$, where the p-value is derived from a hypergeometric test. With a threshold of $P_{score} = \alpha$ all significant networks (i.e., $P_{score} \geq \alpha$) that are associated with desired functions can then be parsed further into the lookup table. For brevity, this work focused on three main functions: **cell death and survival**, **invasion of the cell**, and **proliferation of cell**.

The lookup table is a data matrix that contains information about genes and their functional evidence as contained in the Ingenuity Knowledge Base. For each *significant* gene, the IPA analysis output provides information on how it affects the desired functions if there are any evidence from the literature. It indicates, for instance, whether the gene increase or decrease the function. These findings are abstracted into the lookup table.

Table 3 illustrates a snippet of the lookup table that indicate functional evidence concerning 10 genes. For each *GeneID* the table indicates how it affects any of the three main hallmarks. Four indicator values (i.e., Increases = 1, Decreases = -1, Affects = 0, and No Evidence = NA) were employed to denote functional evidence. A value of zero (0) indicates that there was not much evidence to support the direction to which the gene effects the function, while the value *NA* indicates that there was no evidence to support association of the gene with the function.

From the table, the gene `BMP2` decreases cell invasion as well as survival, while it has no evidence of association to cell proliferation. On the other hand, there is enough evidence that `VEGFA` increases cell invasion and proliferation; it decreases survival of the cell as well. The information contained in the lookup table provides several subjective avenues to construct

prior rules and/or augment rule induction on the target dataset.

Table 3: A snippet of a functional lookup table

	Functional Evidence		
GeneID	**Invasion**	**Survival**	**Proliferation**
ACTA2	1	NA	NA
AKAP12	NA	-1	0
BMP2	-1	-1	NA
CCL4	NA	1	0
COL1A1	NA	-1	1
DUSP6	NA	1	1
VEGFA	1	-1	1
SERPINA3	NA	-1	NA
POSTN	1	NA	1
FN1	1	-1	1

3.2.4 Prior Rules Generation

Unlike MS-TRL, where prior rules are generated from multiple related source rule models, KARL's prior rules generation engine induce prior rules based off information contained in the lookup table. As KARL was designed to augment rule induction with *subjective interestingness* based on particular domain knowledge, which is dependent on the interest of the investigator, prior rule generation can take a diverse turn. Different questions can be asked of the lookup table, which can in turn lead to different prior rules.

The space of prior rules depends on the number of variables contained in the lookup table, size of their values, and size of the class (response) variable values. Prior rules can be restricted to specific domain knowledge or it can be generalized to include every variable contained in the lookup table.

A prior rule of the form: `IF (COL4A2 = UP) THEN (Class = CASE)`, for instance, could be induced if the lookup table suggests that when a particular gene, `COL4A2`, is up-regulated

and there is functional evidence of association to proliferation of cell, then there is a higher likelihood of cancer. This could be based off the investigator's prior knowledge of the domain or test specific hypothesis.

On the other hand, the investigator could induce prior rules with all possible combinations of genes and class values followed by pruning with less stringent statistics. Assume, for example, that a variable contained in the table, `GENE`, takes two values `Down` and `Up`, while the target class variable, `Class`, also take two values, `Case`, and `Control`. Then, prior rules can be instantiated with a permutation of all variable values into the following list of rules:

```
IF (GENE = Down) THEN (Class = Case)
IF (GENE = Up)   THEN (Class = Case)
IF (GENE = Down) THEN (Class = Control)
IF (GENE = Up)   THEN (Class = Control)
```

This mechanism is applied to every variable contained in the lookup table. Thus, between the two scenarios there is a lot of wiggle room for exploration. For brevity, KARL adopted the latter approach to generate prior rules, and the statistic **coverage** (< 5) was used to filter out "bad rules".

3.2.5 Induction of Final Rule Model

The KARL framework was implemented as a semi-automated algorithm as illustrated in fig. 9. The first two components of the framework, the extraction of domain knowledge from multiple sources and its subsequent abstraction into a functional lookup table, are implemented outside the JAVA-based TRL toolkit. A potential future study could consider automating the whole suite by seamlessly integrating the first two components to the rest.

The algorithm accept as inputs the target dataset (D_T), a user specified constraints, (C), for learning good rules, and a lookup table (LKP_{TABLE}) that encodes domain knowledge in a machine-readable csv file. The algorithm first proceeds by preprocessing (e.g., discretization via **EBD**) the target dataset (see—line 5, fig. 9). Second, it applies the LookUpTable2RulesGenerator subroutine (line 6, fig. 9), a special prior rules generation engine,

1: **function** KARL(D_T, LKP_{TABLE}, C)
2: ▷ D_T : the target dataset
3: ▷ LKP_{TABLE} : lookup table contains information culled from multiple sources
4: ▷ C : user specified constraints for rule learning
5: PREPROCESS(D_T)
6: $priorRules \leftarrow$ LookUpTable2RulesGenerator(LKP_{TABLE})
7: $model \leftarrow$ TRL($priorRules, D_T, C$)
8: **return** *model*
9: **end function**

Figure 9: Pseudocode for KARL

to generate prior rules from information contained in the lookup table. Here, the basis for inducing interesting prior rules can take diverse forms as described in section 3.2.4. Finally, with the generated prior rules, processed target dataset, and the user specified constraints, the algorithm applies the modified TRL (see section 3.3) algorithm to induce the final rule model.

3.3 MS-TRL

3.3.1 Background

The fundamental idea of MS-TRL is to learn prior rules from multiple models as opposed to one—a concept akin to seeking advice from multiple experts. Figure 10 illustrates the general configuration for the framework, while fig. 11 represents the implementation algorithm, and explained below.

Given an input of source(s) and target datasets, including user specified constraints to guide rule induction, the algorithm outputs a classification rule model for the target dataset.

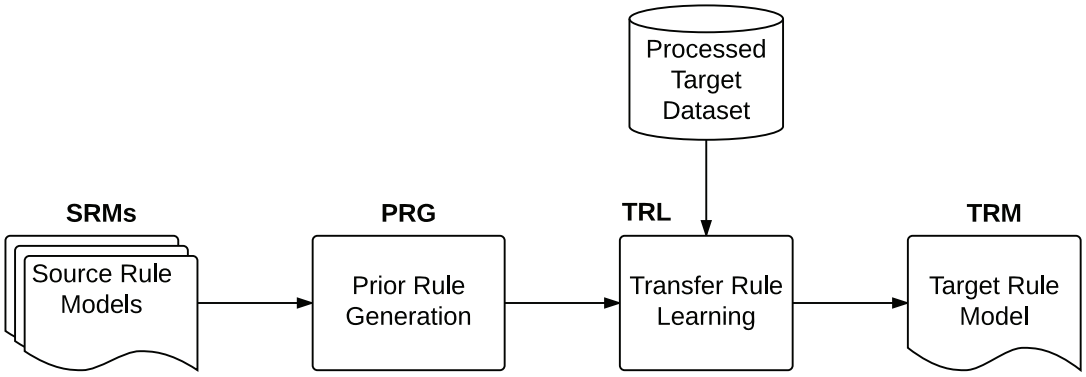

Figure 10: An illustration of the MS-TRL framework

1: **function** MS–TRL($D_{S_{1\ldots N}}, D_T, C$)
2: ▷ $D_{S_{1\ldots N}}$: a set of N source datasets
3: ▷ D_T : target dataset
4: ▷ C : user specified constraints for rule learning
5: PREPROCESS($D_{S_{1\ldots N}}, D_T$)
6: $priorRules \leftarrow \emptyset$
7: **for each** $D_{S_i} \in D_{S_1} \ldots D_{S_N}$ **do**
8: $priorRules \leftarrow priorRules \cup \text{RL}(D_{Si}, C)$
9: **end for**
10: $model \leftarrow \text{TRL}(priorRules, D_T, C)$
11: **return** $model$
12: **end function**

Figure 11: A pseudocode for implementing the MS-TRL framework

The main contributions—algorithmic and conceptual—of this method are: (1) an expansion of the input parameters to accept multiple source datasets, (2) a refined preprocessing sub-

routine module to address previous challenges of transferring *whole rules* between source and target, and (3) a new sub-routine for generating prior rules. Thus, with an input of N source datasets (D_{S_N}), a target dataset (D_T), and user specified constraints for *good rules*, the framework induces a rule model via four main steps: (1) preprocessing, (2) induction of multiple source rule models (SRMs), (3) generation of prior rules from the SRMs, and (4) induction of a target rule model using TRL on the prior rules and target dataset.

3.3.2 Data Preprocessing

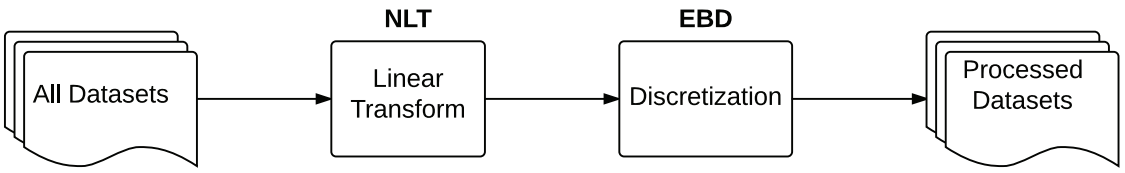

Figure 12: Preprocessing steps for all datasets prior to rule modeling. **NLT**: Normalized Linear Transform; **EBD**: Efficient Bayesian Discretization

The PREPROCESS engine (see line 5, fig. 11 and fig. 12) involves two main steps. First, it applies the *normalized linear transform* (NLT) [97] method to transform the numerical ranges of source variables to that of their corresponding variables in the target. Assume the array matrices for the source and target are denoted, respectively, by $X(n \times p)$ and $Y(n \times q)$, where n is the number of variables, while p and q are their respective sample sizes. The fundamental idea of NLT is to map each sample of the source and target to $AX_i + c$ and $BY_j + d$, such that the numerical range of values for each variable across are identical. Here A and B are transform matrices $(n \times n)$, while a and b are bias vectors. Therefore, for each variable i, you only need to estimate the parameters $\{a_{ii}, c_i\}$ and $\{b_{ii}, d_i\}$ for their respective equations $\{a_{ii}x_{ij} + c_i\}_{j=1}^{p}$ and $\{b_{ii}x_{ij} + d_i\}_{j=1}^{q}$ so that:

$$\min_j(a_{ii}x_{ij} + c_i) = \min_j(b_{ii}y_{ij} + d_i)$$

$$\max_j(a_{ii}x_{ij} + c_i) = \max_j(b_{ii}y_{ij} + d_i)$$

where x_{ij} and y_{ij} denote the values of variable i within the jth sample.

This preprocessing step has several advantages. First, the NLT method preserves the relative ranking order of the expression values for each variable without loss of information. Second, by transforming the source and target variables onto identical numerical range ensures an effective transfer of discretization cut points. Third, it facilitates transfer across disparate assay platforms. Last, it also facilitates transfer via the *whole rule* approach, which has been shown to improve knowledge transfer [23], without much fuss.

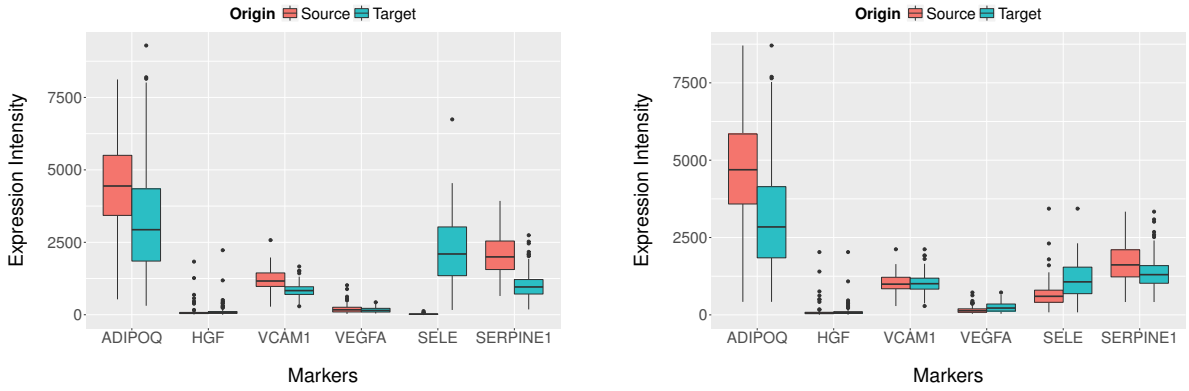

Figure 13: A comparison of the numerical ranges of expression values for example source and target variables via NLT transform. Left: Before NLT transform. Right: After NLT transform.

Figure 13 involves two box-plots that illustrates the distribution of expression intensities for six variables across example source and target datasets. The left and right plots represent, respectively, the intensity distributions before and after the *NLT Transform*. Let us consider the variable `ADIPOQ`. Before the transform, its minimum values within the source and target were respectively 532.1 and 310.0; its maximum values, on the other hand, were respectively, 8121.0 and 9292.0. After the transform, it assumes an identical numerical range between the source and target—that is, identical minimum and maximum values of 421.05 and 8706.05, respectively.

Meanwhile, observe the variable `SELE`. Its numerical range (i.e., $[0.3, 126.5]$) within the source data is way off that of the target (i.e., $[165.0, 6742]$). With such a scenario, transferring discretization cutoff points between the source and target gets quite challenging. Thus, the

NLT scheme facilitates the categorization of source and target variables values (say, `HIGH`), so that they can be readily comparable.

The final preprocessing step involves discretization. First, using **EBD** [141], variables of the target data set are discretized. Then their respective discretization cut points are transferred to their counterparts within the sources. For instance, if the values of `ADIPOQ` are categorized in to groups, `LOW` and `HIGH`, and the cut point is, say, 4350.67, then this value is set as the discretization boundary of `ADIPOQ` across all data sets. That is, within each source and target dataset all values of the variable `ADIPOQ` that are below 4350.67 are categorized as `LOW` (i.e., $-inf \ldots 4350.67 \leftarrow$ `LOW`). Similarly, all values that are at least 4350.67 are denoted as `HIGH` (i.e., $4350.67 \ldots inf \leftarrow$ `HIGH`).

3.3.3 Prior Rules Generation

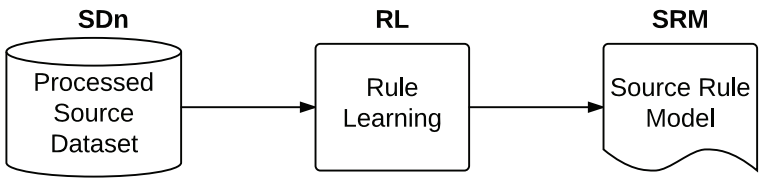

Figure 14: A framework for generating a single source rule model

After the preprocessing phase MS-TRL proceeds with the induction of source rule models (SRMs) on each processed source data set, using RL with the same user specified criteria for learning good rules—fig. 14 illustrates the induction of an SRM. Subsequently, all the SRMs are merged into a unified list of prior rules—see lines 6 to 9 of fig. 11—that together forms an ensemble of prior classification rule models. In addition, the prior rules generation engine ensures that redundant rules are filtered out. When a rule co-occur in multiple source models, counts for its statistics (i.e., **TP** and **FP**) are merged. This strategy provides significant improvements to TRL. First, it improves the confidence of prior rules. Second, it ameliorates *catastrophic forgetting* by "remembering" the performance of rule patterns, which have been discovered in order models. Last, it is able to facilitate the discovery of

domain-specific, as well as domain-independent, rule patterns across multiple data sets.

Furthermore, to identify the source of each prior rule in the final model, the prior rules generation engine annotates each source rule with the name of the source dataset from which it emanated. Finally, the ensemble of prior rule models are used to seed the induction of the final rule model on the target dataset (D_T).

3.3.4 Induction of Final Rule Model

Having seeded the *beam* with the prior rules, induction of the final rule model proceeds as a heuristic beam-search in the space of rules, via a general-to-specific approach (see sections 2.5 and 3.1), using a modified TRL (see the sub-routine on line 10 of fig. 11). A user can opt to specialize only the prior (**OnlyPriors search**) rules during the beam search, or learn and specialize new rules in combination with prior rules (**Combo search**). This option can enable you to test and explore the significant merits of transfer learning. For example, "is there any significant difference in learning performance between specializing on what you already know as opposed to combining new knowledge with what you already know?" is a plausible hypothesis to test.

Meanwhile, with the former—that is, *OnlyPriors search*—individual rule statistics (i.e., **coverage**, **TP**, **FP**) are first updated based on the training data. The rules that do not meet the good-rule criteria are pruned away. The algorithm then proceeds with the usual beam search (i.e., specialize and test) and outputs the content of the beam when none of the rules can be specialized. The heuristic beam search with the latter scheme, however, can be complicated due to potential conflicts between new and prior rules in the search space as they can cover the same data instance.

Several measures were put in place to address these conflicts. First, new rules (and their specializations) that would lead to prior rules, which are already on the beam, are pruned and their respective rule statistics are updated. Second, RL employs inductive strengthening, π, to minimize over-fitting by ensuring that newly induced rules must cover at least π previously uncovered training data instances. This means that rules, which are highly ranked on the beam, can potentially displace other good rules from the final rule list, if they cover the

same data instance(s). Since the objective of MS-TRL was to induce robust and more general rules, rules—especially prior rules that have covered many data—with relatively high coverage must be favored in such conflicts. To that end, we set the **CF** function to *Laplace Estimate*, which penalizes rules with low coverage, as opposed to *PPV*, which bias the **CF** in favor of rules with zero **FP**. Third, prior rules which meet the "good-rule" criteria, have high coverage, but relatively low **CF** to new rule(s) are kept, and included to the final rule list. Future study can device a new **CF** heuristic that will appropriately trade-off coverage and performance.

3.4 iTRL

3.4.1 Background

TRL, TRL-FM, MS-TRL, and KARL are all special cases of incremental learning, where prior knowledge, abstracted as prior rules, is relied upon to potentially increase the learning experience of a target model. With the availability of multiple related datasets integrative modeling can be implemented in an iterative (or on-line) fashion where a rule model learned on particular dataset(s) can be used to seed learning on subsequent models should new data become available. This scenario can develop into a generalized framework for incremental transfer rule learning given multiple datasets. Note that here, the source of the prior rules can emanate from the output of an RL, TRL, TRL-FM, MS-TRL, or KARL models. I've coined this generalized framework *iTRL*—incremental transfer rule learning for multiple related datasets.

The utility of *iTRL* is particularly useful in several instances of integrative modeling of related gene expression data. First, in a scenario of federated modeling, where all the datasets are not available at a time the *iTRL* approach can be used in an *on-line* fashion to learn a model at a time. Second, it can be used as a tool to investigate the role ordering plays in on-line transfer rule learning. The ordering of prior rules can influence the performance and structure of the final rule model. Last, it can be an effective tool to detect robust rule

patterns, as rules that are retained and propagated along intermediate prior models can be detected.

3.4.2 iTRL: The framework

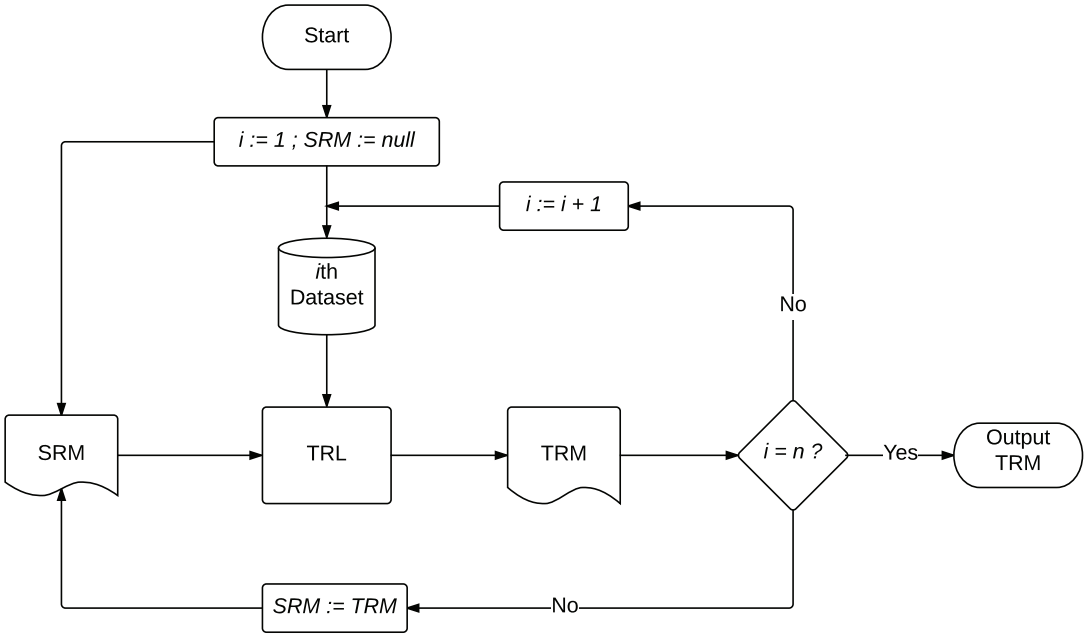

Figure 15: An illustration of the iTRL framework: **SRM**, Source Rule Model; **TRL**, Transfer Rule Learning subroutine; **TRM**, Target Rule Model

Figure 15 is a flowchart that illustrates the *iTRL* framework, while fig. 16 is the implementation algorithm. Unlike MS-TRL, where prior rules are generated from a union of multiple $SRMs$, iTRL generates prior rules from one source model at a time for integrative modeling via incremental transfer learning on multiple related datasets. Note that, for brevity, fig. 15 iterates over multiple datasets using the output of TRL as the source. Future work can increase the complexity by integrating and using KARL, MS-TRL, or even TRL-FM for prior rules generation.

The framework implements a simple feedback loop over TRL on each dataset as follows. Given a set of N datasets, including a user specified constraints for good rules; all the datasets

```
1: function iTRL(D_{1...N}, C)
2:     ▷ D_{1...N} : a set of N datasets
3:     ▷ D_N : is designated as target dataset
4:     ▷ C : user specified constraints for rule learning
5:     PREPROCESS(D_{1...N})
6:     model ← ∅
7:     for i ← 1 to N do
8:         model ← TRL(model, D_i, C)
9:     end for
10:    return model
11: end function
```

Figure 16: Pseudocode for iTRL

are first preprocessed using the same scheme as illustrated in fig. 12. Then, it initializes the SRM (i.e., a set of prior rules) and the loop-counter, i, to null and 1, respectively. It proceeds further by inducing a target rule model, TRM, on the ith dataset using the TRL algorithm. If $i < N$, it is incremented by one and the TRM is set as the new SRM—prior rules—and the loop continues to induce a new TRM on the $(i+1)$th dataset, D_{i+1}, and SRM. On the other hand, if $i = N$ the algorithm outputs TRM. Note that here, the target dataset is always designated as the last (i.e., D_N)

4.0 EXPERIMENTS

The rationale for developing these frameworks was to improve the performance of predictive rule modeling for disease diagnosis, prognosis, and screening. The proposed frameworks are supposed to leverage the advent of biomedical data explosion to improve the discovery of potent insights that could lead to the development of new therapies and/or treatments. The pertinent questions you may ask are how would they improve the status quo; how would they improve learning; what novel biological insights could they provide?

To answer these questions, I performed series of experiments to validate (or otherwise) the usefulness of the proposed frameworks. Note that TRL-FM is a published work, whose key findings motivated the development of KARL, MS-TRL, and iTRL. Its experimental design is thus slightly different from the rest. Section 4.1 briefly describes the experimental design for TRL-FM, while sections 4.2, 4.3 and 4.5 are dedicated for the rest.

4.1 TRL-FM: EXPERIMENTAL DESIGN

Table 60 in appendix C provides details of the three example datasets that we used for the experiments. Each example contained 7 microarray studies of two-group comparison (i.e., case vs control). The datasets were collected from three studies: a brain cancer study, a prostate cancer study, and an IPF study. These datasets particularly suit the goals of our experiments and the utility of integrative modeling of multiple gene expression datasets because, (1) they are publicly available, (2) they have been used extensively to test experiments in several integrative modeling studies, and (3) they were generated using diverse microarray platforms. Testing the flexibility of TRL-FM with datasets generated using diverse platforms

is essential since other methods (e.g., TRL, or meta-analysis) require identical platforms and variables for integrative modeling. That is, TRL-FM avoids the critical and often challenging task of mapping features (e.g., gene names) across disparate platforms for integrative modeling.

To evaluate the feasibility of the TRL-FM framework, we compared its area under the ROC curve (AUC) with models developed with RL (baseline), TRL, and selected machine learning methods namely, Support Vector Machines (SVM), Linear Discriminant Analysis (LDA), Random Forest (RF), C4.5, Naive Bayes (NB), and Penalized Logistic Regression (PLR). In addition, we compared TRL-FM with other integrative models driven by meta-analysis and cross-platform data merging. Using these integrative methods, disease specific datasets were merged into a single matrix for classification modeling.

For meta-analysis, we applied the AW Fisher method [86], while we adopted COMBAT [94], a batch-effect removal method, for cross-platform data merging. The advantage of the AW method over others (e.g., Fisher, Stouffer) is that it is able to weight the relative contributions of each study towards evidence aggregation and elucidates heterogeneity in the analysis. However, while several methods for cross-platform data merging have been proposed, the choice of COMBAT was arbitrary. Tables 61 and 62 (see appendix C) illustrate the data characteristics, while we used the above-named methods to integrate disease-specific datasets (i.e., all datasets that correspond to a particular disease, say IPF) into a single matrix.

4.2 DATA SETS

To evaluate the plausibility of the methods, gene expression data sets were downloaded from Gene the Expression Omnibus [2]. In all, the total number of data sets were 25, and consisted of 5 each from 5 different cancer types (i.e., prostate cancer, brain cancer, breast cancer, lung cancer, and colorectal cancer). Table 4 describes the characteristics of the datasets. For brevity, the following inclusion/exclusion criteria were considered:

1. A data set should contain 30 or more samples.

Table 4: Description of gene expression datasets that were used for KARL, MS-TRL, and iTRL experiments. **Series ID**=GEO accession number, **#S**=number of samples, **T**=tumor samples, **N**=normal samples, **#V**=number of predictor variables

Disease	**Series ID**	**Platform**	**#S (T/N)**	**#V**	**Source**
Brain Cancer	GSE4412	HG-U133A,B	85 (59/26)	28168	Freije et al [158]
	GSE4271	HG-U133A,B	100 (76/24)	28168	Phillips et al [159]
	GSE4290	HG-U133 Plus 2	100 (81/19)	20185	Sun et al [160]
	GSE1993	HG-U133A	58 (39/19)	12501	Petalidis et al [161]
	GSE16011	HG-U133 Plus 2	175 (159/16)	17332	Gravendeel et al [162]
Breast Cancer	GSE15852	HG-U133A	86 (43/43)	12501	Pau et al [163]
	GSE42568	HG-U133 Plus 2	121 (104/17)	20156	Clarke et al [164]
	GSE29431	HG-U133 Plus 2	66 (54/12)	20156	Lopez et al [165]
	GSE7904	HG-U133 Plus 2	62 (43/19)	20156	Richardson et al [166]
	GSE10780	HG-U133 Plus 2	185 (42/143)	20156	Chen et al [109]
Colon Cancer	GSE24514	HG-U133A	49 (34/15)	12501	Alhopuro et al [167]
	GSE23878	HG-U133 Plus 2	59 (35/24)	20185	Uddin et al [168]
	GSE20916	HG-U133 Plus 2	70 (36/34)	20185	Skrzypyczak [169]
	GSE10715	HG-U133 Plus 2	30 (19/11)	20156	Galamb et al [170]
	GSE9348	HG-U133 Plus 2	82 (70/12)	20185	Hong et al [171]
Lung Cancer	GSE7670	HG-U133A	66 (39/27)	12501	Su et al [171]
	GSE10072	HG-U133A	107 (58/49)	12501	Landi et al [172]
	GSE18842	HG-U133 Plus 2	91 (46/45)	20156	Palencia et al [173]
	GSE19188	HG-U133 Plus 2	156 (91/65)	20156	Hou et al [174]
	GSE19804	HG-U133 Plus 2	120 (60/60)	20156	Lu et al [175]
Prostate Cancer	GSE6956	HG-U133A_2	89 (69/20)	12501	Wallace et al [176]
	GSE17951	HG-U133 Plus 2	137 (68/69)	20185	Jia et al [177]
	GSE32448	HG-U133 Plus 2	80 (40/40)	20156	Derosa et al [178]
	GSE46602	HG-U133 Plus 2	50 (36/14)	20156	Mortensen et al
	GSE82188	HG-U133A	136 (65/71)	12501	Wang et al [179]

2. Each data set should contain samples from both normal and primary tumor tissue. For instance, for disease type of lung cancer, search keywords values like "lung cancer" *OR* "adenocarcinoma" and "normal" *OR* "control" *OR* "healthy" for disease and normal samples, respectively, were used to retrieve desired samples.
3. Each data set should contain at least 10 samples for each class category (e.g., control samples). This was a necessary criterion for stratified cross-validation tests. It ensured that each fold would contain at least one sample from each class category.
4. To ensure coverage of relative large pool of variables (i.e., genes), only studies that used Affymetrix Human Genome U133A,B (GPL96, GPL97) and Affymetrix Human Genome U133 Plus 2.0 (GPL570, GPL571, GPL8542) platforms were considered.

4.3 GENERAL EXPERIMENTAL DESIGN

To effectively compare and contrast the effects of learning predictive rule models with the frameworks developed herein, as opposed to RL and TRL, I applied various experimental strategies on the algorithms using the datasets as inputs. Table 5 summarizes the various experimental strategies, including highlights of the algorithm used, input datasets, and the purpose for the experiments. These experiments were motivated by various hypotheses that this work sought to test.

The overarching hypothesis that "predictive rule modeling of gene expression data via multi-source transfer learning improves learning performance" could be decomposed further into two main sub-hypotheses. The first is to test whether learning improves with knowledge transfer. The second, on the other hand, is to ascertain whether knowledge transfer from multiple sources leads to better learning than transfer from a single source. To test these hypotheses effectively it was essential to define baseline—"control"—models.

Table 5: A summary of general experiments

Method	**Datasets**	**Purpose**
RL	All datasets	To determine baseline performance of learning without transfer
TRL	All datasets per cancer type	To determine baseline performance of transfer learning with a single source
KARL + only prior	All datasets	To determine the effects of knowledge augmented rule learning with **only** prior rules
KARL + combo search	All datasets	To determine the effects of knowledge augmented rule learning with **both** prior and new rules
MS-TRL & iTRL + only prior + intra-transfer	All datasets per cancer type	To determine the effects of multi-source transfer by learning with **only** prior rules
MS-TRL & iTRL + combo search + intra-transfer	All datasets per cancer type	To determine the effects of multi-source transfer by learning with **both** prior and new rules
MS-TRL & iTRL + only prior + inter-transfer	One dataset from each cancer type	To determine the effect of multi-source transfer by learning with **only** prior rules among heterogeneous datasets
MS-TRL & iTRL + combo search + inter-transfer	One dataset from each cancer type	To determine the effect of multi-source transfer by learning with **both** prior and new rules among heterogeneous datasets

4.3.1 Baseline Models

RL was set as the baseline for testing the first hypothesis, while TRL was used for the second. RL experiments were ran on each available dataset. For TRL, the choice of input datasets

was dicey because each disease type (or data set) contained five datasets, but the algorithms requires a source and a target to be specified. To address this challenge, an exhaustive pairing of source-target within a disease set was performed. That is, for each target dataset, i, each of the $n - i$ remaining was, in turn, set as source. This strategy was particularly essential to investigate the characteristics of which sources(s) would likely lead to positive or negative transfer.

Essentially, the same model constraints and learning parameters were used for all models. This was necessary to ensure fair comparisons of baseline models and the proposed ones. Thus, model constraints and learning parameters were set as follows: *minimum conjuncts* = 1; *maximum conjuncts* = 5; *minimum coverage* = 4; *minimum TP* = 0.05; *maximum FP* = 0.10; *minimum CF* = 0.80; *beam width* = 2500; *inductive strengthening* = 1; the inference method for conflict resolution was set to *weighted voting*. To test the significance of induced rules, the *likelihood ratio statistic* [133] was used. A significance level of 99% was set a *good-rule* criterion. In addition, I opted for the *Laplace Estimate* as the **CF** function instead of the default, **PPV**.

The problem with **PPV** ($\frac{TP}{TP+FP}$) is that it is not robust and can over-fit rules with relatively low numbers of TP and FP [19]. The **CF** value, for instance, may change significantly for an extra data coverage if both TP and FP are low. Consider two rules, r_1 and r_2, where none misfires (i.e., $FP_1 = FP_2 = 0$), but the first correctly fires one example (i.e., $TP_1 = 1$), while the second correctly fires 99 examples (i.e., $TP_2 = 99$). In this case, both rules would have a CF value of one (i.e., $CF_1 = CF_2 = 1.0$). However, if they both misfire on a new data instance (i.e., $FP_1 = FP_2 = 1$), then their new CF values become $CF_1 = 0.5$ and $CF_2 = 0.99$, respectively. Even though this problem might be reduced by setting a minimum FP criterion, it becomes more pronounced when sample size increases—especially integrative modeling of multiple data sets—and rule *confidence* is measured by the CF value. The *Laplace Estimate* ($\frac{TP+1}{TP+FP+2}$) addresses this problem by adding two "smoothing" examples, one for each class.

Apart from these "default" model constraints and learning parameters, the proposed frameworks have also individual ancillary parameters. The MS-TRL and KARL frameworks, for instance, can induce a rule model with only prior rules or a combination of prior rules and

new ones. The default learning parameters, including the ancillary ones, are command-line arguments, which can always be changed by the user. In addition, there are variations in the experimental design of the proposed frameworks based on input data and learning parameters. The ensuing sections describe experimental design configurations for the proposed methods.

4.4 PROPOSED MODELS

4.4.1 KARL Models

The motivation for KARL experiments was influenced by three aims. The first and foremost was to test the hypothesis that: "predictive rule modeling with KARL is statistically significantly robust than baseline RL." The second was to seek answers to questions like "does augmenting prior rule generation with domain-knowledge lead to significantly better transfer learning than related model-based priors?" since KARL extracts prior rules from domain-knowledge—which could be culled from a domain experts, literature, and/or knowledge bases—as opposed to related models like MS-TRL. Answering this question could give very important perspectives on when to transfer, particularly when the two sources are available for prior rules generation. Third, was to ascertain whether distinct rule patterns could be discovered across multiple models for related studies (e.g., same cancer type) by augmenting rule learning with the KARL approach. In addition, it was desirable to determine how the *combo* search compares and contrasts with the *only-priors* version, while using KARL.

To meet these aims, KARL models were ran with appropriate input parameters. Relevant domain knowledge, for instance, were extracted, abstracted, and encapsulated into the functional lookup tables. Thus, KARL models were trained on each dataset, including their corresponding functional lookup tables, where available. In addition, multi-source transfer learning with KARL was performed under two main schemes. First, conduct rule space search with only the seeded prior rules—that, specialize only prior rules (see fig. 19). The second scenario performs heuristic beam search with prior rules together with newly induced

rules from target dataset—that is, specialize both prior and new rules (see fig. 18).

4.4.2 MS-TRL Models

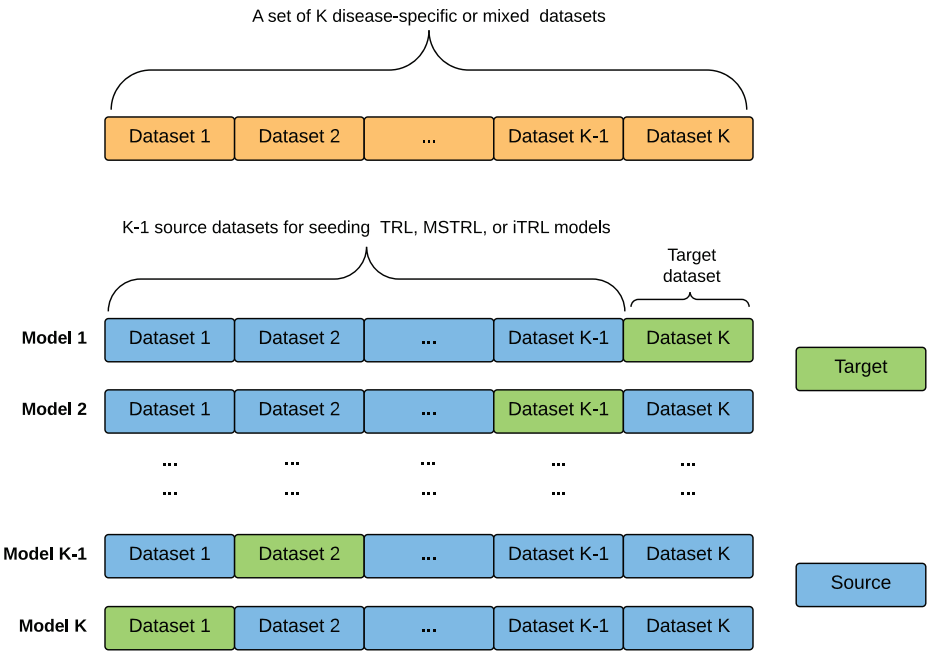

Figure 17: Experimental design for MS-TRL

The purpose of the MS-TRL experiments were motivated by five aims. First, to test the hypothesis that "MS-TRL is more robust than baselines RL and TRL." Second, to investigate the relationship between numbers of sources and positive/negative transfer. That is, does increase in the number of sources correlated with robustness? Third, to investigate the relative importance/contribution of each source model to the final rule model. Fourth, to investigate how MS-TRL could be used to discover robust rule patterns from appropriately determined homogeneous and heterogeneous datasets? Last, to investigate how *combo* search compares and contrasts with the *only-priors*, using MS-TRL.

To meet the above aims several experiments were performed using the MS-TRL framework. Figure 17 illustrates the design for a general experimental set-up. It accepts as input a set of K datasets, which can consist of disease-specific (i.e., same cancer type) datasets

or a mix of disease types (see brown blocks of fig. 17). What is more, the datasets can be randomly pooled from the various disease-specific sets into a mixture of different diseases. This second option is particularly useful to test the feasibility of MS-TRL on a heterogeneous environment. That is, the ability to learn or glean information from remotely related domains and/or models for transfer learning.

With the input set of datasets, one is designated as the target, while the rest (i.e., $K-1$) are earmarked as sources. In fig. 17 the source and target datasets are annotated by light blue and light green-colored blocks, respectively. In each experiment, up to $K-1$ blocks were used for training source models, while the remaining part was used as target. This was repeated K times such that each data block was designated as target once.

Meanwhile, for the $K-1$ available source datasets, an exhaustive combinatorial approach was used to select every possible—combinations—set between 2 to $K-1$ sources. This strategy was also particularly essential to investigate which particular source models, or combination thereof, were more likely to yield positive/negative transfer. In addition, this strategy was also used to investigate whether there is a significant relationship between the number of source models and positive/negative transfer.

4.4.3 iTRL Models

As stated earlier, the iTRL framework was designed for incremental learning with TRL. It is also akin to on-line learning, where an existing rule model is updated when new datasets become available. According to the iTRL framework, the existing model is used as prior rules to seed learning on the new dataset. Here, the source of the existing model is immaterial; that is, it could be a product of RL, TRL, TRL-FM, KARL, or MS-TRL. Therefore, for brevity, iTRL experiments consisted of prior rules that were generated from TRL—that is, RL plus prior rules.

Like MS-TRL, the iTRL experiments were motivated by several aims. First, to test the hypothesis that, "predictive rule modeling via the iTRL approach leads to statistically significantly better models than baseline RL and/or TRL." Second, which is also the overarching aim, is to investigate whether for incremental transfer rule learning via the iTRL approach,

the ordering of datasets significantly affects learning performance of the final model. Third, to investigate whether the number of sources (i.e., iterations) improve learning performance. Fourth, to investigate the feasibility of how iTRL could be used to learn robust rule patterns from both homogeneous and heterogeneous biomedical datasets. Fifth, to investigate how *combo* search compares and contrasts with the *only-priors*, using iTRL; and last, to investigate how iTRL compares and contrasts with MS-TRL, given the same source(s) and target.

To meet these aims the configuration in fig. 17 was also used. Using a similar combinatorial scheme like MS-TRL (see section 4.4.2), every possible ordering—permutations—of 2 to $K-1$ sources in a set of K datasets were considered. That is, for every kth target dataset in a set of K cancer-specific (or mixed) datasets, there were $\frac{(K-1)!}{(K-3)!} + \frac{(K-1)!}{(K-4)!} +, ..., + \frac{(K-1)!}{1!}$ ordering of possible sources. Meanwhile, like KARL and MS-TRL, two types of beam search were performed. First, transfer learning with a specialization of only prior rules (see fig. 19), and second, heuristic beam search with a combination of both prior and new rules (see fig. 18).

4.4.4 Multi-source Transfer Rule Learning: An Example

Figure 18 illustrates an example of the general-to-specific rule space search that the proposed frameworks employ to induce a classification rule model on the target dataset, given prior rules. The prior rules can be gleaned from multiple related models, like MS-TRL, or abstracted from multiple domain knowledge source, like KARL. In this example, heuristic beam search is performed with a combination of prior rules—annotated with light-blue font color—and new rules—annotated with golden-yellow font color. In addition, the source rule models have been encoded as $S1, S2, \ldots, SN$ to facilitate the identification of prior rules provenance.

First, the *beam* is initialized with the transferred prior rules together with the most general rule pattern for inducing new rules. Subsequently, a specialization operator is applied to each rule. Rules that do not meet the *good rule criteria* are pruned away, while the rest stay on the beam and get specialized in the next iteration. The specialization-pruning iteration continues until no rule can be specialized, and the algorithm stops and outputs the

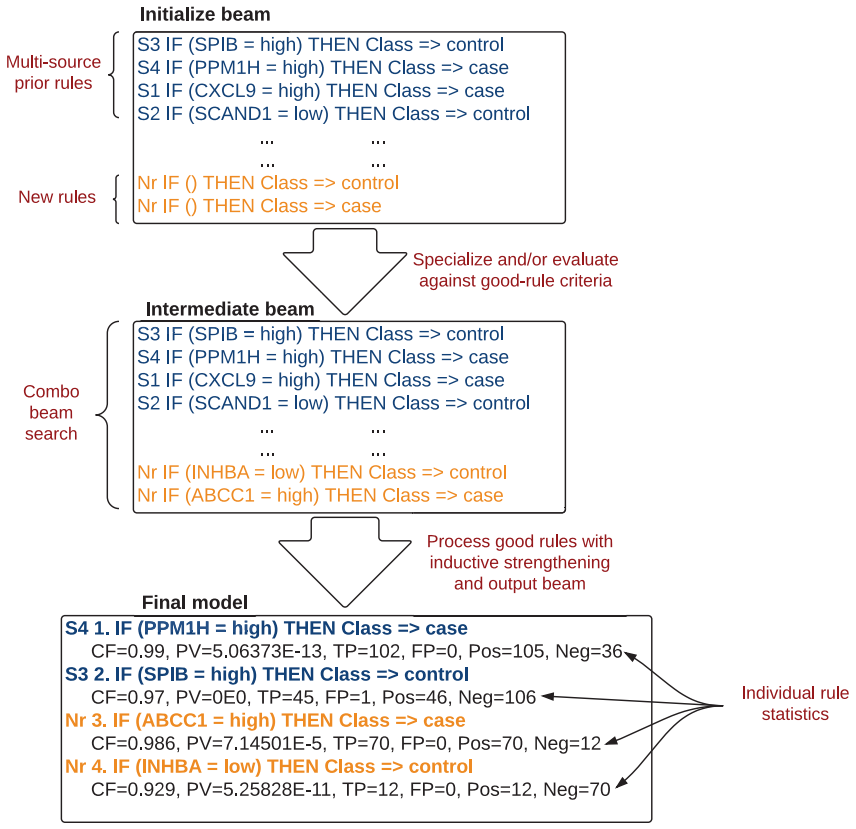

Figure 18: An example of multi-source transfer rule learning

set of rules on the beam. Note that in the first few iterations, specializations (e.g., from the most general rule) of new rules that will potentially lead to prior rules, which are already on the beam, are skipped.

In addition, the final rule model contains rules that were retained from the prior rules. The first two rules, for instance, were retained from source models $S4$ and $S3$, respectively. New rules are annotated with the prefix N_r. Finally, individual rule statistics have also been indicated. The first rule, for instance, has CF, p-value (PV), TP, and FP values of respectively $0.99, 5.06\text{e}{-13}, 100$, and 0. Furthermore, Pos (i.e., 105) indicates the number of training instances that have the same class as this particular rule predicts, while Neg (i.e., 36) indicates the number of training instances that have a different class than it predicts.

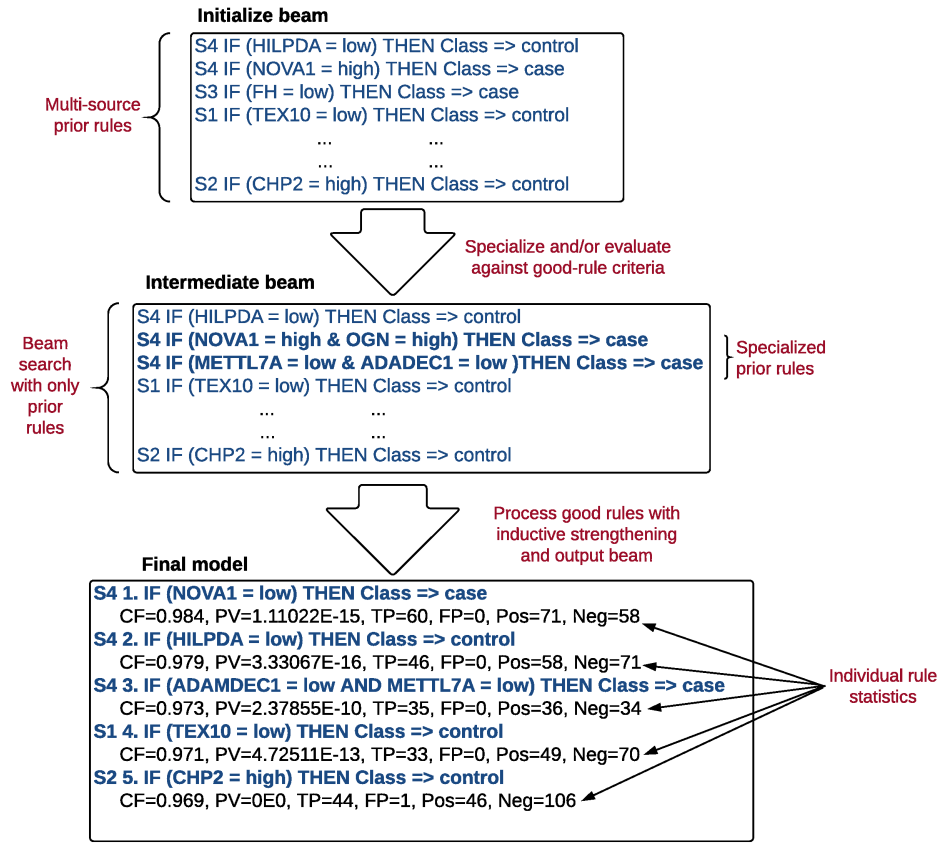

Figure 19: An example of multi-source beam search with only prior rules

Figure 19 depicts an example of prior-rules-only heuristic beam search. That is, it specializes only prior rules where appropriate. Unlike fig. 18, the beam is first initialized with only prior rules gleaned from either multiple related source models or domain knowledge. The search then proceeds with a specialization-pruning iteration until no rule can be specialized and the algorithm stops. Here, the annotation format for the prior rules, including their provenance, are the same as illustrated in fig. 18.

4.5 EVALUATION

The aim of transfer learning is to improve learning on the target task by leveraging information from the source task. *Positive transfer* is said to have occurred if a transfer method improves learning on the target task—in other words, if it improves *completeness* and *consistency* of the target model. As explained in section 2.5, an *ideal* rule learning model is the one which is complete and consistent. The more complete and consistent a rule model is the more robust it is. To test for robustness it was essential to evaluate the rule models that were spawned from the experiments via common measuring instruments.

4.5.1 Metrics

To ascertain whether the proposed frameworks exhibited more completeness and consistency over the baselines several evaluation metrics were employed. While rule coverage statistics were used to evaluate completeness, general classification performance metrics were applied to estimate consistency. Therefore, *positive/negative* transfer can be said to have occurred when there is a significant gain/loss in coverage and classification accuracy by a multi-source transfer rule-learning model over the baseline model.

In addition, other metrics were employed to determine how inherent and related characteristics among the source(s) and target datasets affected learning performance. I present below, in detail, the metrics I used to evaluate gain/loss of completeness and consistency.

1. **Classification**

   To evaluate the classification performance of the models, I estimated the Sensitivity (SN), Specificity (SP), Balanced Accuracy (BACC), and Accuracy (Acc) of baseline and all the multi-source extension models given the same target dataset.

$$SN = \frac{TP}{TP + FP} \qquad SP = \frac{TN}{TN + FN} \qquad BACC = \frac{SN + SP}{2}$$

$$Acc = \frac{TP + TN}{TP + FP + TN + FN}$$

   where, TP (true positive rate) is the fraction of positive examples (i.e., *cases* or *tumor* examples) that were predicted correctly, TN (true negative rate) is the fraction of negative

examples (i.e., *controls* or *normal* examples) that were predicted correctly, FP (false positive rate) is the fraction of negative examples predicted wrongly, and FN (false negative rate) is the fraction of positive examples predicted wrongly.

2. **Coverage**

   Coverage, also known as support, can be used as a measure for the generality and completeness of a rule—the more data examples a rule covers, the more general and complete it is. Coverage can be estimated as the probability that a rule pattern will occur. That is, $p(V_1 \cap ... \cap V_n \cap C)$, where V_i and C are the domain-variable and target class respectively. Several variants of the notion of coverage such coverage difference ($CovDiff = TP - FP$), rate difference ($RateDiff = TP/Pos - FP/Neg$) or positive coverage ($CovPos = TP/Pos$), have been proposed by the inductive learning community. For brevity, I opted for the latter. In addition, I estimated the *CovPos* for each class. That is, $p(V_1 \cap ... \cap V_n | C = CASE)$ for cases and $p(V_1 \cap ... \cap V_n | C = CONTROL)$ for controls. For each cross-validation experiment the minimum, maximum, and median *CovPos* were computed. For brevity, the maximum possible *CovPos* was used due to inductive strengthening—that is, priority was given to rules that cover as much of the training examples.

3. **Abstentions**

   A closely related metric to coverage is abstention. A rule model may abstain from making a prediction if none of it rules cover a test/validation data instance. The number of instances that the model abstains from is referred to as *Abstentions*. While coverage can indicate the completeness of a rule model given set of training examples, the rate of Abstentions could be indicative of both its completeness and consistency since it evaluates coverage on the *unseen* test example set. The more a rule abstains from making a prediction the less complete it is. Thus, abstentions can also be used to estimate *positive/negative* transfer. Here, positive transfer is said to occur if there is significance difference in abstentions between baseline and proposed model.

4. **Significance difference**

   To determine that the general learning performances (i.e., positive/negative transfer) between baseline and proposed frameworks did not occur by chance, test of significance

difference were requirement. To that end, I employed the *Paired t-test* and the *Wilcoxon Signed-rank* tests at significance level of $\alpha = 0.05$.

5. **Level of relatedness**

   Mutual information (**MI**) was used to evaluate the relationship between relatedness and direction of transfer (i.e., positive/negative). MI is the measure of the amount of information shared by data sets [180]. Given two random variables X and Y with a joint probability mass function $p(x, y)$ and, respectively, marginal probability mass functions of $p(x)$ and $p(y)$, the MI can be estimated as:

$$MI\,(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 \left(\frac{p(x,y)}{p(x)p(y)} \right) \tag{4.1}$$

   In cases were X, Y, or both are continuous, estimating MI can be very tricky due to difficulty in estimating the underlying probability density function. Several methods (e.g., binning) have been proposed to convert continuous variables to discrete ones before estimating the MI. Most of the "binning" based methods, however, have inherent drawbacks—for example, no optimal way of estimating bin size—, so I adopted a "nearest neighbors" method, which was first proposed by [181]. This method has also been shown to be robust [182] for estimating MI between a discrete and continuous variable. Thus given a dataset, you can use MI to quantify the extent to which a disease state (discrete variable) affects the expression value (continuous) of a given gene. Meanwhile, MI between X and Y can range between zero (i.e., no relation or they are independent) and ∞. Any MI value higher than 0 is indicative of relatedness, so it is essential to standardize the upper bound. The global correlation coefficient (λ), a standardized measure of MI [183], is given by the formula:

$$\lambda(X,Y) = \{1 - \exp[-2MI(X,Y)]\}^{1/2} \tag{4.2}$$

   where $0 \leq \lambda \leq 1$. Using a subjective threshold (i.e., 0.1), I used λ to identify domain-specific genes. That is, for a given dataset, a gene, X, is said to be a disease-specific (or otherwise disease-dependent) variable, if $\lambda(X,Y) > 0.1$, where Y denote the disease state (i.e., cancer or normal). It is expected that the more domains (datasets in this

case) are related the more domain-independent variables they share. In other words, the degree of relatedness between source(s) and target is related to the degree of independent variables they share. This assumption was necessary to investigate the notion that: "the more source and target datasets are related, the more transfer improves target task."

6. **Robust patterns**

   A rule is said to be robust if it occurs in at least 50% of source models within a model. Thus for MS-TRL, a robust pattern is a final rule that was retained from at least 3 sources. For the case of iTRL a pattern is robust when it has "survived" (traversed) at least 3 iterations. This metric was particularly useful in identifying domain-specific as well pan-cancer patterns. That is, when a robust pattern occurs within a disease set it can be termed domain-specific; it is general or domain-independent (pan-domain) when it spans multiple domains.

4.5.2 Cross-Validation

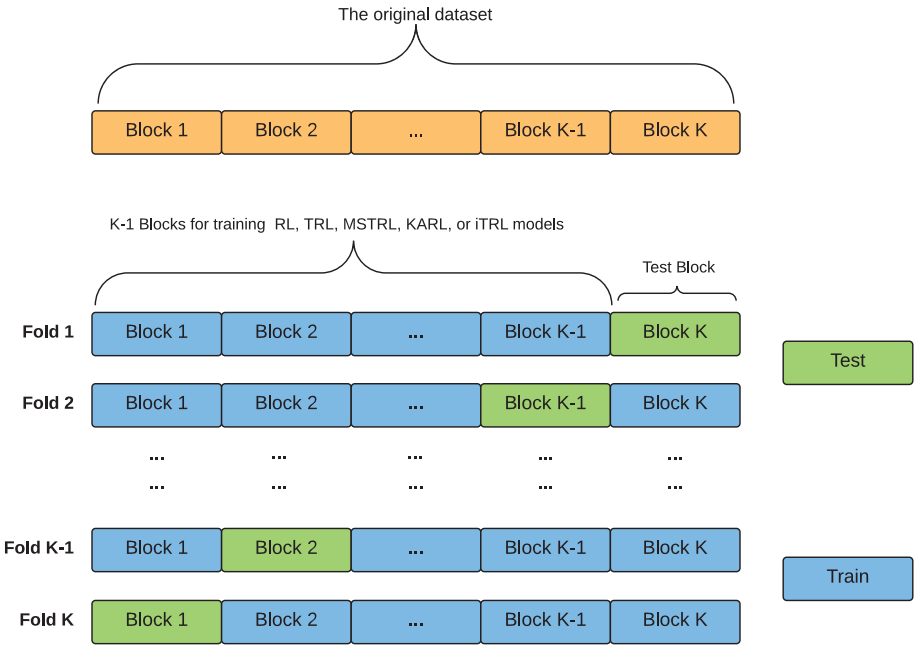

Figure 20: K-fold Cross Validation

To reduce over-fitting of the frameworks, as well as estimate robust classification performance, K-Fold cross-validation scheme was used. Each data set was partitioned into K blocks, while ensuring that the distribution of class variable was balanced in each block. In each experiment, K-1 blocks were used for training a model (i.e., baseline or proposed), while the remaining part was used for testing. This was repeated K times such that each block was used for testing once. Finally, the predictive performances (i.e., sensitivity, specificity, and balanced accuracy) for all K experiments were averaged. Figure 20 illustrates a schematic depiction of the K-fold cross-validation. For this work, K was set to 10 for all cross-validation experiments.

5.0 RESULTS & ANALYSIS

This chapter presents and analyzes the results of the experiments that were described in the previous chapter. Similarly, section 5.1 briefly presents and discusses results for the published work, TRL-FM. Further, section 5.2 describes the performances for the baseline models—RL and TRL, while sections 5.3 to 5.5, discuss KARL, MS-TRL, and iTRL, respectively.

5.1 TRL-FM: RESULTS & DISCUSSION

TRL-FM was able to transfer interpretable classification rules via abstracted biological knowledge in the form of FMs. Most of the abstracted FMs were associated with known hallmarks of cancer [151]. In addition, knowledge transfer via an ensemble of FMs, more often than not, performed better than individual FMs. This observation corroborated results from other studies which have reported that a combination of FMs(e.g., group of pathways) improves performance for integrative analysis of genomic data [152, 153].

Meanwhile, after comparing TRL-FM with other methods, results (see tables 6 to 8) show that TRL-FM statistically significantly outperforms TRL as well as other traditional models based on single source data. Furthermore, TRL-FM performed better than other integrative models driven by meta-analysis and cross-platform merging.

In summary, the capability of utilizing transferred abstract knowledge derived from source data using feature mapping enables the TRL-FM framework to mimic the human process of learning and adaptation when performing related tasks. This enables the framework to intelligently incorporate domain knowledge that traditional methods might disregard, to boost predictive power and generalization performance. In this study, TRL-FM's abstrac-

Table 6: AUC comparison of TRL-FM with SVM, LDA, RF, C4.5, NB, and PLR on all datasets. Note that for TRL, the AUC for the highest performing source is shown, while for TRL-FM, the medium of knowledge transfer is the union of all FMs.

Dataset	**SVM**	**LDA**	**RF**	**C4.5**	**NB**	**PLR**	**RL**	**TRL**	**TRL-FM**
Emblom	1.00	1.00	1.00	0.98	0.96	0.98	0.97	0.97	0.94
Freije	0.74	0.72	0.72	0.73	0.82	0.76	0.76	0.78	0.80
Gravendeel	0.52	0.59	0.59	0.53	0.63	0.56	0.49	0.49	0.59
KangA	0.93	0.86	0.86	0.79	0.94	0.90	0.86	0.93	0.97
KangB	0.91	0.87	0.87	0.87	0.91	0.95	0.83	0.91	0.95
Konishi	0.90	0.68	0.68	0.74	0.90	0.90	0.78	0.83	0.95
Lapointe	0.96	0.91	0.91	0.94	0.97	0.96	0.93	0.93	0.97
Larsson	0.33	0.67	0.67	0.58	0.67	0.67	0.75	0.75	1.00
Nanni	0.70	0.61	0.61	0.44	0.57	0.65	0.54	0.54	0.64
Pardo	0.83	0.85	0.85	0.63	0.80	0.88	0.85	0.90	0.95
Paugh	0.48	0.45	0.45	0.43	0.50	0.45	0.51	0.52	0.54
Petalidis	0.75	0.71	0.71	0.69	0.80	0.80	0.83	0.88	0.91
Phillips	0.73	0.70	0.70	0.66	0.75	0.80	0.66	0.73	0.78
Singh	0.89	0.90	0.90	0.89	0.88	0.91	0.89	0.89	0.93
Sun	0.59	0.66	0.66	0.63	0.70	0.73	0.73	0.73	0.84
Varambally	1.00	0.92	0.92	0.67	1.00	1.00	0.83	1.00	1.00
Wallace	0.82	0.85	0.85	0.76	0.81	0.87	0.76	0.81	0.84
Welsh	0.94	0.66	0.66	0.79	0.93	0.94	0.92	0.95	0.93
Yamanaka	0.57	0.57	0.57	0.56	0.71	0.56	0.50	0.50	0.79
Yang	0.69	0.51	0.51	0.89	0.57	0.73	0.94	0.94	0.89
Yu	0.94	0.93	0.93	0.80	0.97	0.94	0.88	0.90	0.93
AVG AUC	0.77	0.74	0.74	0.71	0.80	0.81	0.77	0.80	0.86
AVG SEM	0.06	0.07	0.07	0.07	0.06	0.05	0.07	0.06	0.04

Table 7: This table shows the average classification performance per disease type as compared to merged datasets per disease type. In the dataset column, **Avg** denotes average, **MM** denotes merged by meta-analysis, and **M** means merged by cross-platform data merging. * denotes that transfer learning methods were not evaluated. Currently, TRL and TRL-FM cannot be applied to cross-domain studies (i.e., transfer from one disease type to another).

Dataset	**SVM**	**LDA**	**RF**	**C4.5**	**NB**	**PLR**	**RL**	**TRL**	**TRL-FM**
Average performance per disease type									
Avg_Brain	0.67	0.66	0.66	0.64	0.73	0.69	0.66	0.68	0.76
Avg_IPF	0.80	0.78	0.78	0.78	0.82	0.86	0.85	0.89	0.95
Avg_Prostate	0.89	0.83	0.83	0.76	0.88	0.90	0.82	0.86	0.89
Merged per disease type by meta-analysis									
MM_Brain	0.67	0.70	0.70	0.69	0.70	0.69	0.67	*	*
MM_IPF	0.88	0.88	0.88	0.85	0.74	0.88	0.81	*	*
MM_Prostate	0.89	0.84	0.84	0.81	0.70	0.85	0.76	*	*
Merged per disease type by batch effect removal									
M_Brain	0.50	0.51	0.51	0.48	0.53	0.51	0.54	*	*
M_IPF	0.67	0.63	0.63	0.60	0.63	0.64	0.68	*	*
M_Prostate	0.53	0.53	0.53	0.53	0.53	0.55	0.59	*	*

Table 8: A Mann-Whitney paired-sample signed rank test with significance level $\alpha = 5\%$. P-values were adjusted with the Benjamini Hochberg method. Significant p-values are displayed in bold font.

Method	**SVM**	**LDA**	**RF**	**C4.5**	**NB**	**PLR**	**RL**	**TRL**
LDA	0.1230							
RF	0.1230							
C4.5	**0.0386**	0.0943	0.0943					
NB	0.3453	**0.0076**	**0.0076**	**0.0035**				
PLR	0.0737	**0.0043**	**0.0043**	**0.0043**	0.6280			
RL	0.3473	0.6825	0.6825	**0.0137**	0.1151	0.0700		
TRL	0.6924	0.0648	0.0648	**0.0017**	0.8666	0.6825	**0.0076**	
TRL-FM	**0.0094**	**0.0006**	**0.0006**	**0.0002**	**0.0094**	**0.0217**	**0.0017**	**0.0052**

tion of knowledge is achieved in the form of functional modules, but the overall framework is generalizable in that different approaches of acquiring abstract knowledge can be integrated into this framework.

5.2 BASELINES

5.2.1 Classification performance - RL

Tables 9 and 10 represent cross-validation (10 fold) results for RL on all data sets. The table rows have been partitioned into five blocks, each representing a set of a type of cancer, i.e., *brain*, *breast*, *colon*, *lung*, and *prostate* in that order. For each dataset the mean Sensitivity (**SN**), Specificity (**SP**), Balanced Accuracy (**BACC**), Accuracy (**Acc**), and Abstentions (**Ab**) under each cross-validation experiment have been shown. In addition, we recalculated the accuracies by considering an abstention as error. This is also presented as Accuracy

Table 9: Mean classification (cross-validation) performance of baseline RL, using weighted-voting for inference. **SN** = Sensitivity, **SP** = Specificity, **BACC** = Balanced Accuracy, **Acc** = Accuracy, **AccAb** = Accuracy including abstentions, **Ab** = Abstentions

Dataset	**SN**	**SP**	**BACC**	**Acc**	**AccAb**	**Ab**	**Ab (%)**
GEO16011	98.73	0.00	49.37	90.17	89.14	2	1.14
GEO1993	92.31	73.68	83.00	86.21	86.21	0	0.00
GEO4271	100.00	8.33	54.17	78.00	78.00	0	0.00
GEO4290	100.00	10.53	55.26	82.83	82.00	1	1.00
GEO4412	94.83	53.85	74.34	82.14	81.18	1	1.18
GEO10780	42.50	100.00	71.25	87.43	86.49	2	1.08
GEO15852	86.05	83.72	84.88	84.88	84.88	0	0.00
GEO29431	100.00	100.00	100.00	100.00	93.94	4	6.06
GEO42568	100.00	81.25	90.63	97.48	95.87	2	1.65
GEO7904	92.86	47.37	70.11	78.69	77.42	1	1.61
GEO10715	83.33	37.50	60.42	69.23	60.00	4	13.33
GEO20916	100.00	97.06	98.53	98.55	97.14	1	1.43
GEO23878	100.00	100.00	100.00	100.00	94.92	3	5.09
GEO24514	100.00	80.00	90.00	93.88	93.88	0	0.00
GEO9348	100.00	83.33	91.67	97.50	95.12	2	2.44
GEO10072	100.00	95.83	97.92	98.10	96.26	2	1.87
GEO18842	100.00	90.48	95.24	95.40	91.21	4	4.40
GEO19188	96.70	92.31	94.51	94.87	94.87	0	0.00
GEO19804	94.92	95.00	94.96	94.96	94.17	1	0.83
GEO7670	94.87	87.50	91.19	92.06	87.88	3	4.55
GEO17951	85.08	87.69	86.38	86.36	83.21	5	3.65
GEO32448	97.22	90.00	93.61	93.42	88.75	4	5.00
GEO46602	97.22	91.67	94.44	95.83	92.00	2	4.00
GEO6956	98.53	55.00	76.77	88.64	87.64	1	1.12
GEO82188	85.94	92.75	89.35	89.47	87.50	3	2.21

Table 10: Mean classification (cross-validation) performance of baseline RL, using minimum p-value for inference. **SN** = Sensitivity, **SP** = Specificity, **BACC** = Balanced Accuracy, **Acc** = Accuracy, **AccAb** = Accuracy including abstentions, **Ab** = Abstentions

Dataset	**SN**	**SP**	**BACC**	**Acc**	**AccAb**	**Ab**	**Ab (%)**
GEO16011	82.28	40.00	61.14	78.61	77.71	2	1.14
GEO1993	74.36	84.21	79.29	77.59	77.59	0	0.00
GEO4271	78.95	45.83	62.39	71.00	71.00	0	0.00
GEO4290	88.75	47.37	68.06	80.81	80.00	1	1.00
GEO4412	77.59	88.46	83.02	80.95	80.00	1	1.18
GEO10780	82.50	93.01	87.75	90.71	89.73	2	1.08
GEO15852	74.42	86.05	80.23	80.23	80.23	0	0.00
GEO29431	100.00	100.00	100.00	100.00	93.94	4	6.06
GEO42568	94.18	81.25	87.71	92.44	90.91	2	1.65
GEO7904	59.52	73.68	66.60	63.93	62.90	1	1.61
GEO10715	50.00	62.50	56.25	53.85	46.67	4	13.33
GEO20916	100.00	100.00	100.00	100.00	98.57	1	1.43
GEO23878	100.00	100.00	100.00	100.00	94.92	3	5.09
GEO24514	88.24	86.67	87.45	87.76	87.76	0	0.00
GEO9348	98.53	100.00	99.27	98.75	96.34	2	2.44
GEO10072	98.25	95.83	97.04	97.14	95.33	2	1.87
GEO18842	97.78	92.86	95.32	95.40	91.21	4	4.40
GEO19188	96.70	87.69	92.20	92.95	92.95	0	0.00
GEO19804	93.22	91.67	92.44	92.44	91.67	1	0.83
GEO7670	82.05	100.00	91.03	88.89	84.85	3	4.55
GEO17951	94.03	80.00	87.02	87.12	83.94	5	3.65
GEO32448	86.11	90.00	88.06	88.16	83.75	4	5.00
GEO46602	97.22	91.67	94.44	95.83	92.00	2	4.00
GEO6956	77.94	65.00	71.47	75.00	74.16	1	1.12
GEO82188	84.38	94.20	89.29	89.47	87.50	3	2.21

including abstentions (**AccAb**).

Generally, RL performs fairly well on all cancer data sets, except poor specificity from

the brain cancer set (i.e., an average of 29.28% and 61.17% for tables 9 and 10, respectively) that was largely due to skewness and inherent heterogeneity within the datasets. The lung cancer set recorded the best baseline performance (i.e., mean **SN** and **SP** of 97.30% and 92.22%, respectively as shown, for instance, from table 9). Thus, the baseline classification performance could be influenced by inherent characteristics of the datasets.

Meanwhile, total abstentions varied across the disease sets. Brain and prostate cancer sets, respectively, recorded the least(4) and most(15) total abstentions. The rate of abstention can be dependent on both the inherent properties of input datasets and generalizability of the induced rule model.

For table 9 the inference method for conflict resolution was evaluated by weighted-voting (default), while the minimum p-value approach was used for the results in table 10. The *weighted-voting* method, which is dependent on the **CF**, is set-up to particularly bias towards rules that maximize **TP** and minimize **FP**. This means that for relatively highly skewed data majority of the induced rules will cover the majority class, and would results in low sensitivity or specificity. See, for example, the performances for datasets *GEO16011*, *GEO4271*, and *GEO10780*, which respectively, have *case-control* class distributions of 159/16, 76/24, and 42/143. As the results show from table 9 and fig. 21, these datasets have low specificity and sensitivity due to their inherent skewness and the choice of inference method for resolving conflicts.

The *minimum p-value* inference method, which is also based on the likelihood ratio statistic, on the other hand, relies on the class distribution of the training set to determine the reliability and/or significance of an induced rule. Thus,it would likely not penalize sensitivity/specificity for highly skewed data sets. In fact, it trades-off sensitivity and specificity for majority of skewed data sets. In like manner, see the results for same datasets in table 10 and fig. 21, where there was a significant improvement in specificity (i.e., *GEO16011*, *GEO4271*) and slight loss in sensitivity. In general, the *minimum p-value* inference method for baseline RL improves balanced accuracy for datasets that are highly skewed.

Figure 21 illustrates how inference via weighted-voting (left) compares with that of minimum p-value (right) over the space of classification performance. The *x-axis* denotes the number of case (or control) examples as a percentage of the total sample size, while perfor-

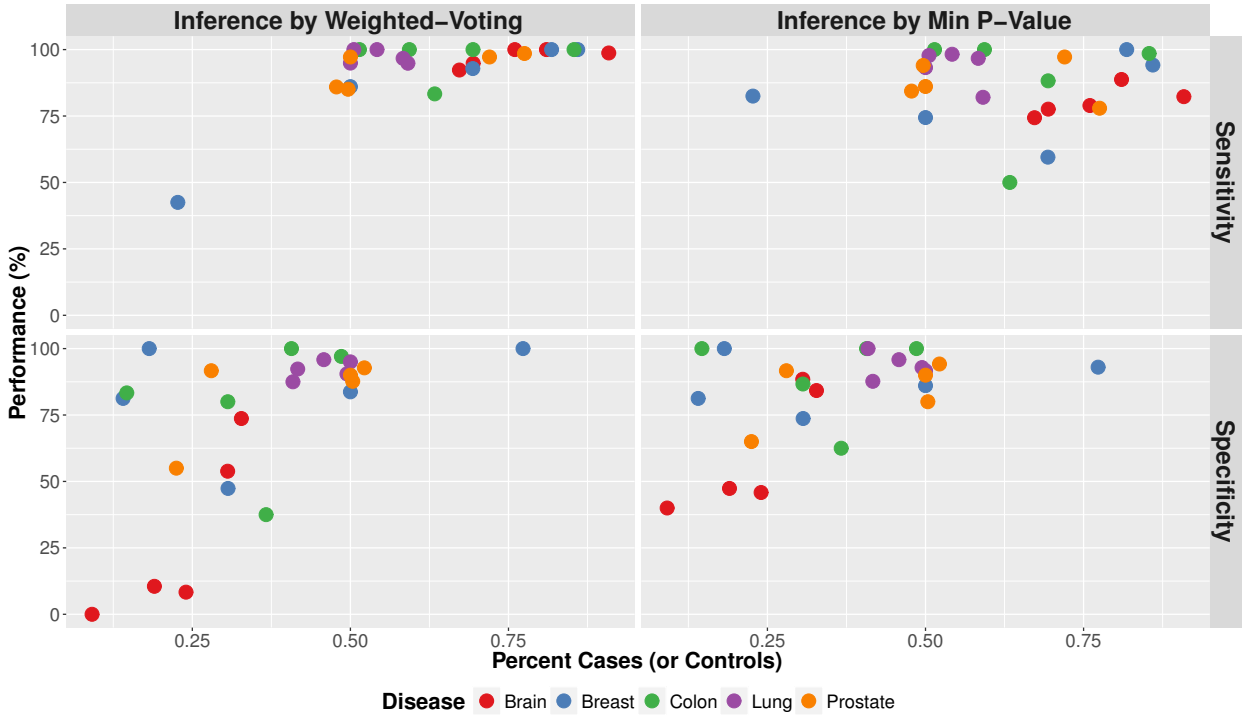

Figure 21: Performance by inference via Weighted-Voting vs Min P-Value

mance in sensitivity(top) and specificity (bottom) are presented on the *y-axis*.

In general, and as discussed earlier, inference via weighted-voting is highly sensitive (i.e., $sensitivity > 75\%$) for datasets which have 50% or more *tumor* examples—see top left. On the other hand, specificity for some datasets, particularly brain cancer, which have 40% or less *normal* examples—see bottom left—have relatively low specificity (i.e., $specificity < 55\%$). Meanwhile, in general, inference via minimum p-value trades-off slightly lower sensitivity (see top right) for relatively higher specificity (bottom right). A future inference method that could fuse and balance these two is desirable. This could be particularly useful for both the *MS-TRL* and *iTRL* frameworks were evidence of robust rules are encapsulated in prior rules statistics such as coverage. For brevity, the weighted-voting method was maintained for the rest of the frameworks.

5.2.2 Coverage - RL

Tables 11 and 12 describe the positive coverage statistics of baseline RL on 10-fold cross-validation. Each table presents the minimum, maximum, and median positive coverage for both case and control examples. That is **ConMin** (minimum positive coverage for controls), **ConMax** (maximum positive coverage for controls), **ConMdn** (median positive coverage for controls), **CaseMin** (minimum positive coverage for cases), **CaseMax** (maximum positive coverage for cases), and **CaseMdn** (median positive coverage for cases). While table 11 presents the coverage per cancer type, table 12 provides a detailed RL coverage for all datasets.

Table 11: The average coverage of baseline RL per cancer type on 10-fold cross-validation. **ConMin** = minimum positive coverage for controls, **ConMax** = maximum positive coverage for controls, **ConMdn** = median positive coverage for controls, **CaseMin** = minimum positive coverage for cases, **CaseMax** = maximum positive coverage for cases, **CaseMdn** = median positive coverage for cases

Disease	**ConMin**	**ConMax**	**ConMdn**	**CaseMin**	**CaseMax**	**CaseMdn**
Brain	0.482	0.673	0.556	0.534	0.765	0.621
Breast	0.652	0.835	0.744	0.660	0.810	0.721
Colon	0.961	0.970	0.965	0.968	0.982	0.975
Lung	0.797	0.932	0.869	0.865	0.952	0.922
Prostate	0.535	0.766	0.671	0.602	0.782	0.678

In general, the brain cancer set presented the lowest (i.e., **ConMdn** = 0.556 and **CaseMdn** = 0.621) baseline coverage for both cases and controls, while the colon cancer set recorded the highest (i.e., **ConMdn** = 0.965 and **CaseMdn** = 0.975). The low coverage within the brain cancer set, particularly among the controls, could be a likely cause of its relatively low specificity. For the prostate cancer set, the relatively lower coverage—second to brain cancer—could also be associated to its relatively higher abstentions. Meanwhile, apart from presenting the highest average coverage, the colon cancer set also shows the least variation (i.e., almost zero) in average coverage. This could also be attributed to the reasons why it recorded one of the best classification performances (see tables 9 and 10).

Table 12: Mean positive coverage for baseline RL models on 10-Fold cross-validation. **ConMin** = minimum positive coverage for controls, **ConMax** = maximum positive coverage for controls, **ConMdn** = median positive coverage for controls, **CaseMin** = minimum positive coverage for cases, **CaseMax** = maximum positive coverage for cases, **CaseMdn** = median positive coverage for cases

Dataset	**ConMin**	**ConMax**	**ConMdn**	**CaseMin**	**CaseMax**	**CaseMdn**
GEO16011	0.513	0.688	0.596	0.549	0.790	0.643
GEO1993	0.603	0.737	0.675	0.607	0.735	0.655
GEO4271	0.426	0.580	0.503	0.491	0.820	0.631
GEO4290	0.433	0.544	0.462	0.519	0.745	0.586
GEO4412	0.436	0.817	0.544	0.503	0.736	0.588
GEO10780	0.497	0.836	0.649	0.402	0.601	0.495
GEO15852	0.421	0.847	0.637	0.362	0.723	0.488
GEO29431	1.000	1.000	1.000	1.000	1.000	1.000
GEO42568	0.889	0.902	0.895	0.975	0.981	0.978
GEO7904	0.451	0.591	0.538	0.561	0.747	0.642
GEO10715	0.869	0.909	0.889	0.895	0.953	0.924
GEO20916	1.000	1.000	1.000	1.000	1.000	1.000
GEO23878	1.000	1.000	1.000	1.000	1.000	1.000
GEO24514	0.934	0.941	0.937	0.944	0.958	0.951
GEO9348	1.000	1.000	1.000	1.000	1.000	1.000
GEO10072	1.000	1.000	1.000	1.000	1.000	1.000
GEO18842	1.000	1.000	1.000	1.000	1.000	1.000
GEO19188	0.576	0.869	0.686	0.740	0.956	0.907
GEO19804	0.583	0.904	0.789	0.652	0.828	0.749
GEO7670	0.824	0.885	0.871	0.935	0.974	0.954
GEO17951	0.346	0.583	0.466	0.536	0.812	0.618
GEO32448	0.419	0.817	0.689	0.439	0.633	0.519
GEO46602	1.000	1.000	1.000	1.000	1.000	1.000
GEO6956	0.556	0.700	0.642	0.639	0.820	0.694
GEO82188	0.354	0.728	0.558	0.395	0.643	0.557

5.2.3 Classification performance - TRL

Tables 13 and 14 represent classification performance of TRL on brain and *mixed* cancer set, respectively. Recall that each dataset within each disease set was designated as target, while the rest, individually, in turn, served as source. In all, there were 21 TRL experiments involving each disease set. For each experiment, the results present the names of the source, target, and measured performance such as sensitivity (SN), specificity (SP), balanced accuracy (BACC), accuracy where abstentions are considered as errors, and percentage of abstentions.

Table 13: Classification performance of TRL Combo on brain cancer set

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**	**Ab (%)**
GEO4290	GEO16011	98.734	0.000	49.367	89.143	1.143
GEO4412	GEO16011	98.734	0.000	49.367	89.143	1.143
GEO1993	GEO16011	98.113	12.500	55.307	90.286	0.000
GEO4271	GEO16011	96.855	6.250	51.553	88.571	0.000
GEO4412	GEO1993	92.308	73.684	82.996	86.207	0.000
GEO4271	GEO1993	89.744	73.684	81.714	84.483	0.000
GEO4290	GEO1993	97.436	68.421	82.928	87.931	0.000
GEO16011	GEO1993	97.436	68.421	82.928	87.931	0.000
GEO16011	GEO4271	100.000	12.500	56.250	79.000	0.000
GEO4412	GEO4271	100.000	20.833	60.417	81.000	0.000
GEO4290	GEO4271	100.000	16.667	58.333	80.000	0.000
GEO1993	GEO4271	100.000	16.667	58.333	80.000	0.000
GEO1993	GEO4290	98.765	31.579	65.172	86.000	0.000
GEO4412	GEO4290	95.062	42.105	68.583	85.000	0.000
GEO16011	GEO4290	100.000	10.526	55.263	82.000	1.000
GEO4271	GEO4290	96.296	21.053	58.674	82.000	0.000
GEO4290	GEO4412	98.305	50.000	74.153	83.529	0.000
GEO4271	GEO4412	93.220	53.846	73.533	81.176	0.000
GEO1993	GEO4412	89.831	73.077	81.454	84.706	0.000
GEO16011	GEO4412	94.915	42.308	68.611	78.824	0.000

Note these results were produced from a version of TRL with combination (Combo) rule

Table 14: Classification performance of TRL Combo on mixed cancer set

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**	**Ab (%)**
GEO7904	GEO4412	94.915	53.846	74.381	82.353	0.000
GEO7670	GEO4412	93.103	53.846	73.475	80.000	1.176
GEO6956	GEO4412	98.305	57.692	77.999	85.882	0.000
GEO9348	GEO4412	93.103	53.846	73.475	80.000	1.176
GEO7904	GEO6956	98.551	55.000	76.775	88.764	0.000
GEO9348	GEO6956	95.588	55.000	75.294	85.393	1.124
GEO7670	GEO6956	95.588	55.000	75.294	85.393	1.124
GEO4412	GEO6956	95.588	55.000	75.294	85.393	1.124
GEO6956	GEO7670	94.872	74.074	84.473	86.364	0.000
GEO4412	GEO7670	92.308	81.481	86.895	87.879	0.000
GEO9348	GEO7670	94.872	88.462	91.667	90.909	1.515
GEO7904	GEO7670	94.872	85.185	90.028	90.909	0.000
GEO6956	GEO7904	95.349	31.579	63.464	75.806	0.000
GEO4412	GEO7904	97.674	42.105	69.890	80.645	0.000
GEO9348	GEO7904	95.349	36.842	66.095	77.419	0.000
GEO7670	GEO7904	97.674	42.105	69.890	80.645	0.000
GEO6956	GEO9348	100.000	83.333	91.667	97.561	0.000
GEO7904	GEO9348	100.000	83.333	91.667	97.561	0.000
GEO7670	GEO9348	100.000	75.000	87.500	96.341	0.000
GEO4412	GEO9348	100.000	75.000	87.500	96.341	0.000

space search; it is akin to MS-TRL with one source, and served as baseline for MS-TRL with two or more sources. For brevity, I will use results for only two disease sets (i.e., brain and mixed) to illustrate the general performance patterns that were observed. Results for the rest of the sets (i.e., breast, colon, lung, and prostate) have been provided as additional material (see appendix B).

In general, positive/negative transfer due to single-source transfer rule learning depends on the individual baseline (i.e., RL) performances of the source and target datasets. For all things being equal, if the source and target have, say, about equal sensitivity, but the

specificity of the source is way higher than that of the target, then positive transfer is most likely to occur. See, for instance, the transfer of GEO1993 to GEO16011 or GEO4412 to GEO4271, all from the brain cancer set (see table 13). In both cases, the individual specificity (see table 9) of the sources were far greater than their respective targets. Similar observations were made on the *mixed* set (see table 14), where transfer from GEO6956 to GEO4412 and GEO9348 to GEO7670. Note that in the case of the latter example, positive transfer occurred due to the relatively higher difference in sensitivity, than specificity, of the source (GEO9348) over that of the target (GEO7670).

Conversely, positive transfer is less likely to occur when individual performance of the target far outweigh that of the sources. In table 13, for instance, transfer from all sources to the target, GEO1993, could not improve performance. Similarly, and as observed from table 14, performance on the target, GEO9348, was either maintained or reduced. These two target, in particular, had relatively higher specificity among their cohorts of their respective disease set.

Finally, TRL with Combo search reduces the rate of abstentions, as observed from both tables 13 and 14. Thus, the utility of the combination search strategy applies to TRL as well. Generally, it increases the coverage space, and hence the generality, of the rule model.

In addition, tables 15 and 16 represent the classification performance of TRL on the same brain and *mixed* set. Note that the search strategy used here was the "OnlyPriors" approach. Similarly, this is akin to MS-TRL OnlyPriors with one source dataset. It was thus used as baseline, as well as basis, for multi-source transfer rule learning with two or more sources. Like the Combo version, and for brevity, results for the rest of the disease sets have been provided as additional material (see appendix B).

As expected, there were quite a lot of abstentions from these experiments, and they affected the general classification by accuracy (see **AccAb** column). Within the subspace of data that it covered (see **BACC** column), it improved performance in most cases, especially when the class distributions within the data are highly skewed (e.g., GEO4290 or GEO16011).

Observe that for cases where the rate of abstentions were low (say, $< 4\%$) it performs well (see transfer from GEO4412 and GEO4290 to GEO4271 in table 15, and GEO7670

Table 15: Classification performance of TRL OnlyPriors on brain cancer set

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**	**Ab (%)**
GEO4290	GEO16011	99.194	20.000	59.597	70.857	26.286
GEO4412	GEO16011	99.270	0.000	49.635	77.714	20.000
GEO1993	GEO16011	95.205	50.000	72.603	83.429	8.571
GEO4271	GEO16011	90.667	54.545	72.606	81.143	8.000
GEO4412	GEO1993	89.474	50.000	69.737	74.138	3.448
GEO4271	GEO1993	86.111	68.421	77.266	75.862	5.172
GEO4290	GEO1993	97.297	29.412	63.355	70.690	6.897
GEO16011	GEO1993	97.368	50.000	73.684	77.586	6.897
GEO16011	GEO4271	97.183	36.842	67.013	76.000	10.000
GEO4412	GEO4271	89.474	62.500	75.987	83.000	0.000
GEO4290	GEO4271	94.667	54.545	74.606	83.000	3.000
GEO1993	GEO4271	90.667	52.381	71.524	79.000	4.000
GEO1993	GEO4290	84.416	57.895	71.155	76.000	4.000
GEO4412	GEO4290	84.211	52.632	68.421	74.000	5.000
GEO16011	GEO4290	95.890	20.000	57.945	73.000	12.000
GEO4271	GEO4290	88.000	53.333	70.667	74.000	10.000
GEO4290	GEO4412	91.071	43.478	67.275	71.765	7.059
GEO4271	GEO4412	96.491	45.833	71.162	77.647	4.706
GEO1993	GEO4412	90.741	68.000	79.370	77.647	7.059
GEO16011	GEO4412	98.214	16.667	57.440	68.235	12.941

to GEO9348 in table 16), and even much better for highly skewed data. Finally, and as discussed above, positive transfer hardly occur when, individually, the target performs much better than the sources.

The TRL results as discussed above would be used in sections 5.4 and 5.5 to compare and contrast multi-source transfer rule learning with MS-TRL and iTRL, respectively, as well as baseline RL. The next section presents results for multi-source transfer rule learning via explicit augmentation of domain knowledge (aka KARL).

Table 16: Classification performance of TRL OnlyPriors on mixed cancer set

Source	Target	SN	SP	BACC	AccAb	Ab (%)
GEO7904	GEO4412	95.745	21.429	58.587	56.471	28.235
GEO7670	GEO4412	94.340	43.750	69.045	67.059	18.824
GEO6956	GEO4412	88.889	62.500	75.694	68.235	17.647
GEO9348	GEO4412	50.000	71.429	60.714	18.824	65.882
GEO7904	GEO6956	96.154	50.000	73.077	35.955	55.056
GEO9348	GEO6956	42.857	87.500	65.179	11.236	83.146
GEO7670	GEO6956	63.636	86.667	75.152	38.202	46.067
GEO4412	GEO6956	86.111	58.333	72.222	42.697	46.067
GEO6956	GEO7670	100.000	66.667	83.333	80.303	9.091
GEO4412	GEO7670	90.625	72.222	81.424	63.636	24.242
GEO9348	GEO7670	81.250	85.714	83.482	57.576	30.303
GEO7904	GEO7670	97.297	91.304	94.301	86.364	9.091
GEO6956	GEO7904	94.118	20.000	57.059	54.839	29.032
GEO4412	GEO7904	96.552	45.455	71.003	53.226	35.484
GEO9348	GEO7904	73.333	42.857	58.095	22.581	64.516
GEO7670	GEO7904	89.744	33.333	61.538	64.516	12.903
GEO6956	GEO9348	98.529	83.333	90.931	93.902	2.439
GEO7904	GEO9348	100.000	90.000	95.000	96.341	2.439
GEO7670	GEO9348	100.000	66.667	83.333	95.122	0.000
GEO4412	GEO9348	100.000	36.364	68.182	90.244	1.220

5.3 KARL

5.3.1 Domain & model variables

Table 17 presents the characteristics of domain and model variables for knowledge augmented rule learning. For each dataset, the variables extracted from its functional lookup table are the number of gene-function associations (**GFA**), and number of functional genes (**FG**). Note that GFA can be significantly larger than FG in some instances. This is because some of the genes are multi-functional. In addition, FG formed the building blocks of **RLT**, the number

Table 17: Characteristics of domain & model variables for KARL models. **GFA** = # of gene-function associations, **FG** = # of functional genes, **RLT** = # of prior rules instantiated from Lookup Table, **PrCo** = # of priors rules in final model via combo search, **TrCo** = total number of rules via combo search, **PvCo** = # of prior variables in model via combo search, **TvCo** = total number of variables in model via combo search, **TvOp** = total number of variables in model via priors only search, **TrOp** = total number of rules in model via priors only search

Dataset	**GFA**	**FG**	**RLT**	**PrCo**	**TrCo**	**PvCo**	**TvCo**	**TvOp**	**TrOp**
GEO16011	42	26	34	9	30	8	29	14	22
GEO1993	81	45	114	11	21	11	21	13	11
GEO4271	133	77	221	13	41	13	41	22	23
GEO4290	100	57	117	11	32	11	32	20	21
GEO4412	157	97	255	24	36	24	36	21	16
GEO10780	2	1	2	0	22	0	21	0	0
GEO15852	27	16	60	7	25	7	25	11	14
GEO29431	726	433	965	2	4	2	4	3	2
GEO42568	819	487	1509	5	6	5	6	8	7
GEO7904	25	15	28	9	21	9	21	10	9
GEO10715	15	10	13	5	10	5	10	5	5
GEO20916	296	176	575	2	4	1	2	2	2
GEO23878	274	170	533	2	4	2	3	3	2
GEO24514	188	113	343	5	8	4	7	5	4
GEO9348	485	297	768	2	4	2	4	3	2
GEO10072	163	98	286	2	2	1	1	2	2
GEO18842	627	381	1057	2	4	1	3	3	2
GEO19188	497	296	969	5	9	4	8	14	9
GEO19804	248	147	539	15	24	13	22	16	9
GEO7670	235	139	439	6	12	6	12	8	5
GEO17951	70	42	146	13	32	13	32	15	14
GEO32448	80	43	122	13	22	13	22	16	14
GEO46602	204	118	326	2	2	1	1	5	3
GEO6956	47	29	48	9	20	9	20	12	11
GEO82188	66	36	128	17	38	13	34	20	16

of good prior rules that were instantiated from the functional lookup table. Furthermore, **PrCo** and **PvCo**, respectively, constitute the number of prior rules and variables that made it to the final rule model using the combination search methods, while **TvOp** and **TrOp**, respectively, represent the number of total variables and rules in the final model via search with only prior rules. Finally, **TvCo** and **TrCo** denote the total number of variables and rules that form the final model via the combination search, respectively.

In general, there were no direct relationship between FG and RLT on the one hand, and PrCo and TrCo on the other. Though some models (e.g., *GEO42568* and *GEO18842*) with quite high RLTs, only a minute proportion (i.e., 5/1509 and 2/1057) of them made it to the final model. Meanwhile, other models with relatively low RLTs (e.g., *GEO7904* and *GEO10715*) had relatively higher proportions of PrCo/RLT (i.e., 9/28 and 5/13, respectively). Thus, a variable might be functionally relevant to a particular domain, but may not necessarily meet the good rules criteria, which is mostly dependent on rule statistics derived from the data.

A relatively large RLT, however, affords the rule learner a wider space from which to choose better and robust rules. This might also lead to the discovery of rules that command stronger support (i.e., coverage) and confidence (i.e., certainty factor); such rules would more likely lead to parsimonious (i.e., few list of rules) rule models. On the contrary, a small FG/RLT may cause a very low PrCo/TrOp. A stark evidence of this claim was identified in *GEO10780*, which had no transferred prior rule. In fact, its FG list contain only one gene and more so the rules instantiated from its RLT were not robust enough to make it to the final model. Thus, the amount of robust prior rules likely to be retained in the final model may not depend solely on the biological relevance of the functional genes, but their total size.

5.3.2 Classification performance - KARL

Table 18 represent the average classification performance (see table 63 in appendix D for entire datasets), while using KARL with combination search (aka Combo) approach. As observed from baseline RL, on the average, brain cancer recorded the worst performance (i.e.,

AccAb = 83.591), while lung (i.e., **AccAb** = 94.788), followed closely by colon (i.e., **AccAb** = 94.273) cancer presented the highest classification performance. The low performance from the brain cancer, especially low specificity, could be attributed to a high degree of inherent heterogeneity. In addition, the rate of abstaining from predicting a new data instance, on the average, ranges from 0.0% (brain and lung cancer) to 0.912% (colon cancer). This trend is not too different from that of baseline RL, albeit the rates are much lower with the *Combo* method.

Table 18: Average classification performance per disease, using KARL Combo method. **SN** = Sensitivity, **SP** = Specificity, **BACC** = Balanced Accuracy, **Acc** = Accuracy, **AccAb** = Accuracy including abstentions, **Ab** = Rate of abstentions

Disease	**SN**	**SP**	**BACC**	**Acc**	**AccAb**	**Ab (%)**
Brain	95.484	37.238	66.362	83.591	83.591	0.000
Breast	83.191	83.030	83.111	89.443	89.254	0.216
Colon	96.667	91.879	94.273	95.046	94.273	0.911
Lung	97.648	91.225	94.436	94.788	94.788	0.000
Prostate	94.033	86.698	90.365	92.092	91.957	0.147

Table 19: Average classification performance per disease, using KARL OnlyPriors method. **SN** = Sensitivity, **SP** = Specificity, **BACC** = Balanced Accuracy, **Acc** = Accuracy, **AccAb** = Accuracy including abstentions, **Ab** = Rate of abstentions

Disease	**SN**	**SP**	**BACC**	**Acc**	**AccAb**	**Ab (%)**
Brain	87.869	59.307	73.588	81.066	78.849	2.787
Breast	95.174	54.106	74.640	76.766	67.349	24.393
Colon	92.500	93.667	93.083	94.492	84.460	11.813
Lung	97.358	93.510	95.434	95.893	89.954	6.193
Prostate	85.223	88.327	86.775	86.653	81.983	6.165

Similarly, table 19 show the average classification performance (see table 64 in appendix D for entire datasets), while using KARL with only prior rules search (aka OnlyPriors) option. Here the average performance ranges from 67.349 (breast cancer) to 89.954 (lung cancer).

Meanwhile, the average rate of abstentions range from 2.787 (brain cancer) to 24.393 (breast cancer). As explained in section 5.3.1, the apparent high average rate of abstentions on the breast cancer set was mainly due to a low number of functional genes, coupled with low support from dataset *GEO10780*.

As we expected, the *Combo* search approach was more accurate, overall (i.e., accuracies with abstentions), than the *OnlyPriors* method. We take the better performance of the former to be due largely to its ability to blend background domain knowledge with newly discovered knowledge into a more robust and expansive rule model. Meanwhile, within the space of examples that the latter covers, it performed quite well. See, for instance, the average BACC for brain, colon, and lung cancer sets, where it outperformed the Combo method. A future method could investigate and device intelligent methods, which can specialize and apply the priors only approach to only specific subsets of data, where it has been determined *a prior* that it would perform well.

5.3.3 Coverage - KARL

Table 20 show the average coverage statistics per disease, using KARL Combo models on 10-fold cross-validation (see table 65 in appendix D for coverage statistics on entire datasets). As baseline RL, the colon cancer set presented, on the average, the highest coverage with medians of 0.950 and 0.965 in normal and tumor examples, respectively. It also showed the least variation in positive coverage—that is $[0.918, 0.971]$ and $[0.916, 0.991]$ for normal and tumor samples, respectively. As expected, brain cancer recorded the worst average coverage (median = 0.556 and 0.616 for controls and cases respectively). This has been a recurring pattern among all the models so far.

Similarly, table 21 represent the average coverage statistics per disease for KARL, using the priors only search option (see table 66 in appendix D for entire datasets). Once again, colon cancer recorded the most average coverage for both normal ($[0.892, 0.938]$) and tumor ($[0.883, 0.889]$) examples. Meanwhile, on the average, the breast and brain cancer sets showed the least coverage (i.e., $[0.480, 0.548]$ and $[0.433, 0.783]$) on normal and tumor samples, respectively. The relative average drop in coverage on the breast cancer set could

Table 20: Average positive coverage per disease, using KARL Combo approach. **ConMin** = minimum positive coverage for controls, **ConMax** = maximum positive coverage for controls, **ConMdn** = median positive coverage for controls, **CaseMin** = minimum positive coverage for cases, **CaseMax** = maximum positive coverage for cases, **CaseMdn** = median positive coverage for cases

Disease	**ConMin**	**ConMax**	**ConMdn**	**CaseMin**	**CaseMax**	**CaseMdn**
Brain	0.386	0.729	0.556	0.463	0.788	0.616
Breast	0.648	0.840	0.759	0.638	0.825	0.719
Colon	0.918	0.971	0.950	0.916	0.991	0.965
Lung	0.763	0.980	0.855	0.829	0.974	0.910
Prostate	0.494	0.876	0.657	0.550	0.874	0.691

Table 21: Average positive coverage per disease, using KARL, prior rules only option. **ConMin** = minimum positive coverage for controls, **ConMax** = maximum positive coverage for controls, **ConMdn** = median positive coverage for controls, **CaseMin** = minimum positive coverage for cases, **CaseMax** = maximum positive coverage for cases, **CaseMdn** = median positive coverage for cases

Disease	**ConMin**	**ConMax**	**ConMdn**	**CaseMin**	**CaseMax**	**CaseMdn**
Brain	0.521	0.727	0.633	0.433	0.783	0.632
Breast	0.480	0.548	0.503	0.634	0.729	0.687
Colon	0.892	0.938	0.914	0.883	0.896	0.889
Lung	0.778	0.855	0.825	0.849	0.958	0.908
Prostate	0.634	0.851	0.744	0.568	0.788	0.691

be due to the reasons as explained in sections 5.3.1 and 5.3.2.

As expected, knowledge augmented rule learning using "prior rules only" does not cover much data examples as the combination method. It does not incorporate new information for knowledge discovery and decision making, therefore restricting itself to a subspace of the entire knowledge space. This is the reason why, comparatively, its rates of abstentions and

coverage are higher and lower, respectively. The next section will provide further comparative analyses of these two approaches against baseline accuracy and coverage—particularly, if they are statistically significantly better.

5.3.4 KARL vs RL

Sections 5.3.5 to 5.3.7 present results for significance tests on whether KARL improves the consistency (i.e., accuracy) as well as completeness (i.e., coverage and abstentions) on the baseline model. Section 5.3.5 reports on the difference in classification performance, while sections 5.3.6 and 5.3.7 recount on differences in coverage and rate of abstentions respectively.

5.3.5 Difference in classification performance

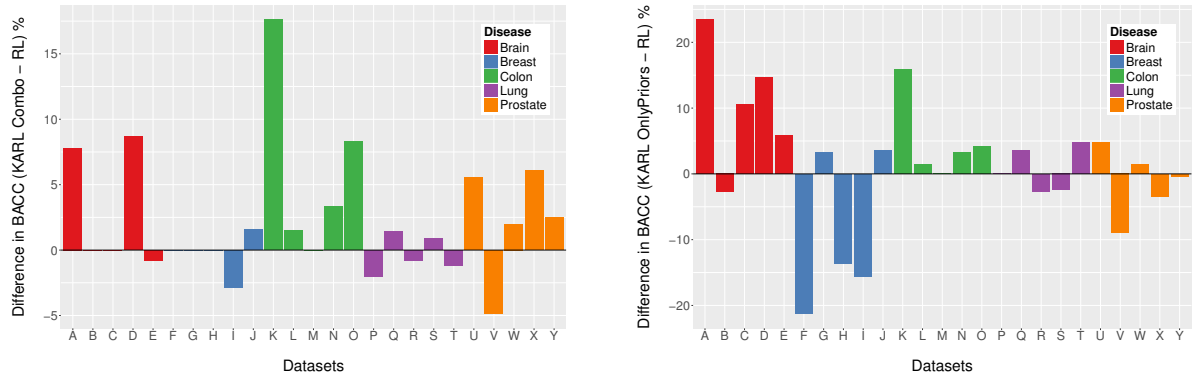

Figure 22: A comparison of classification performance between KARL and RL using balanced accuracy. Left: Difference in BACC - KARL Combo vs RL. Right: Difference in BACC - KARL OnlyPriors vs RL.

Figure 22 displays the difference in classification performances between KARL Combo and RL on the one hand (left) and KARL OnlyPriors (right) on the other, on all datasets. The performance metric used here is BACC. Note that positive bars indicate a gain in performance (i.e., positive transfer), while bars below the *x-axis* depict negative transfer. For data points, where there are no bars (e.g., B: GEO1993, C: GEO4271, or G: GEO15852 from the left figure) the difference in performance was zero—no gain or loss.

The *Wins:Ties:Loses* ratio between KARL Combo and RL was *13:6:6*, while that between the OnlyPriors version and RL was *14:2:9*. In general, KARL Combo made gains on the brain, colon, and prostate cancer sets; the gains were more pronounced in the latter two. The OnlyPriors version, on other hand, made majority gains in the brain and colon cancer sets—especially, the former.

In addition, using a paired t-test and a Wilcoxon signed-rank test, on all datasets, at a significance level of $\alpha = 0.05$, the classification performance of KARL Combo, using BACC, was significantly ($p = 0.0287$ and $p = 0.03624$, respectively) better than baseline RL. Similarly, applying the same significance tests between the OnlyPriors version and RL, the difference in performance was not significant ($p = 0.5454$ and $p = 0.2591$, respectively).

The spikes seen within the brain and colon cancer sets, for both methods, were largely due to gains in specificity. For prostate cancer, however, a mix of gains caused the spikes presented from KARL Combo in both sensitivity and specificity. Thus, gains in performance might depend on the nature of the dataset and the characteristics of the background knowledge that covers it. For instance, KARL OnlyPriors tends to perform well on the control samples from the brain set. It averaged a gain in specificity in excess of 10% on the brain cancer set alone.

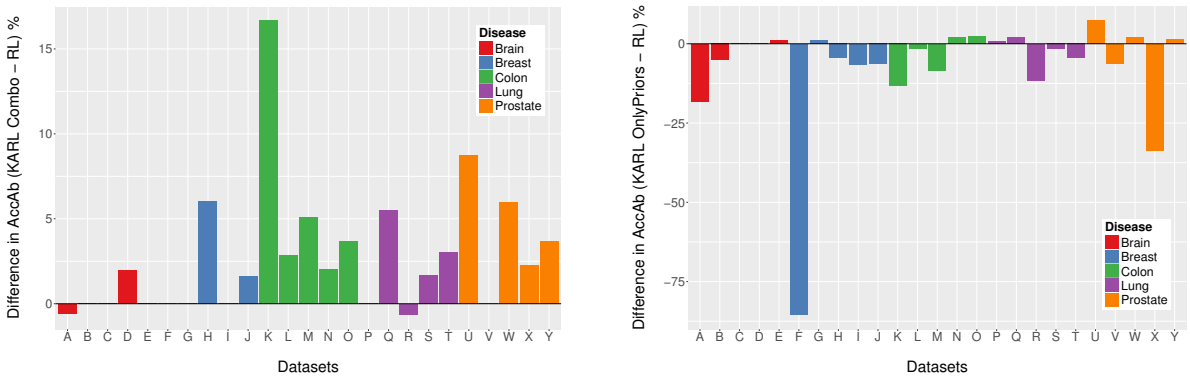

Figure 23: A comparison of classification performance between KARL and RL using accuracy, and penalizing abstentions. Left: Difference in AccAb - KARL Combo vs RL. Right: Difference in AccAb - KARL OnlyPriors vs RL.

Classification performance metrics (like SN and SP), which are based solely on the cov-

erage space of a rule model might not properly elucidate completeness and consistency. A rule model, for instance, which covers a meager 5% of the entire validation examples might command an accuracy of 100% if it gets all the predictions right. Comparing such a model to another that covers, say, 90% of the validation set, and yet scores an accuracy of 80% might not be fair. Recall that an *ideal* rule model must be both complete and consistent. To that end, we penalized abstentions as errors and factored it into the calculation of accuracy—accuracy including abstentions (**AccAb**).

Figure 23 displays the difference in performance, using AccAb, between both the Combo (left) and OnlyPriors (right) versions and baseline RL. Here, we see a significant gain of the Combo version over RL in almost all disease sets. Its *Wins:Ties:Loses* ratio over the baseline are respectively *15:8:2*. The gains were particularly pronounced in the colon cancer set, and followed closed by prostate and lung cancer sets. At a significance level of $\alpha = 0.05$, and using paired t-test and Wilcoxon signed-rank tests, the gains were significant (i.e., $p = 0.001328$ and $p = 0.00102$, respectively). Observe that the gains recorded using AccAb holds more power than BACC; $p = 0.0287$ vs $p = 0.001328$ on paired t-test. The Wilcoxon signed-rank test results was even more powerful (i.e., $p = 0.03624$ vs $p = 0.00102$)

As expected the OnlyPriors version, which predominantly does specialized learning within a subspace of the training set, performed poorly using the AccAb metric. Its *Wins:Ties:Loses* ratio over the baseline was *9:2:14*. While the gains, on the average, were marginal, its loses were relatively substantial. The breast cancer set recorded the worst performance, and the reasons are largely due to abstentions (see section 5.3.7 for an in-depth analysis on abstention differences). The drop in performance on the baseline RL, was marginally significant ($p = 0.05234$ and $p = 0.02166$, respectively on paired t-test and Wilcoxon signed-rank test, all the same level of $\alpha = 0.05$).

5.3.6 Difference in Coverage

Figure 24 represents the difference in maximum positive coverage between KARL and baseline RL for *normal* samples on all datasets. Specifically, the left compares KARL Combo with RL, while the right, on the other hand, compares RL with the OnlyPriors version.

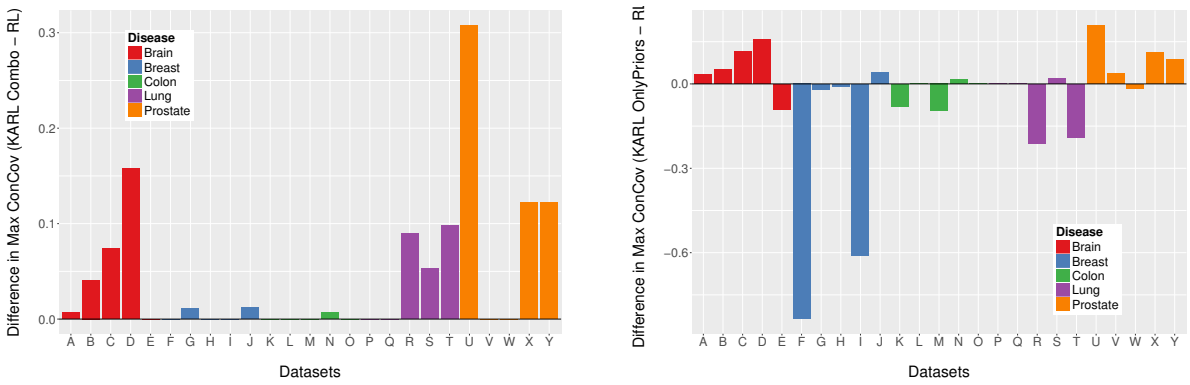

Figure 24: A comparison of maximum positive coverage of control examples between KARL and RL. Left: Difference in Max CovPos of Control Examples - KARL Combo vs RL. Right: Difference in Max CovPos of Control Examples - KARL OnlyPriors vs RL.

The *Wins:Ties:Loses* ratio over that of the baseline was *13:12:0*. Thus, **CovPos** on control training examples of the Combo models was at least that of the baseline. That is, the difference is either zero or more. This observation was not too surprising as the additional background knowledge incorporated into the Combo models allows it to cover at least more training examples. Meanwhile, at a significance level of $\alpha = 0.05$, this gain in coverage over control examples was statistically significant ($p = 0.006069$ and $p = 0.001651$, using paired t-test and Wilcoxon signed-rank test respectively).

With the OnlyPriors method, gains in **CovPos** was more pronounced in the brain and prostate cancer sets; and as expected it suffered the worst loss in the breast cancer set. The *Wins:Ties:Loses* ratio as compared to that of the baseline was *11:4:10*. Using the same level and methods of significance tests as above, the difference was not significant ($p = 0.2675$ and $p = 0.8757$).

Furthermore, fig. 25 shows the difference in maximum **CovPos** between KARL and baseline RL over tumor training examples. Similarly, the left represents the coverage difference between the Combo and RL, while the right depicts that of OnlyPriors and RL. Like the normal examples, the difference in **CovPos** for cases is zero or more for the Combo. The *Wins:Ties:Loses* ratio as compared to the baseline was *9:16:0*. The distribution of the 9

gains were 3, 2, 2, 1, 1 for prostate, brain, colon, and breast cancer, respectively. Expectedly, the general difference of **CovPos** between the Combo and RL over tumor examples was significant ($p = 0.01269$ and $p = 0.009152$).

For the OnlyPriors method, the *Wins:Ties:Loses* ratio to baseline **CovPos** over tumor examples was *10:7:8*. The difference was not significant ($p = 0.4093$ and $p = 1.0$). Finally, the distribution of gains/loss among the disease types are quite random. Thus, in general, the Combo method statistically significantly improves baseline positive coverage on both tumor and normal examples. The OnlyPriors method, on the other hand, improves baseline coverage on specific subsets of training examples.

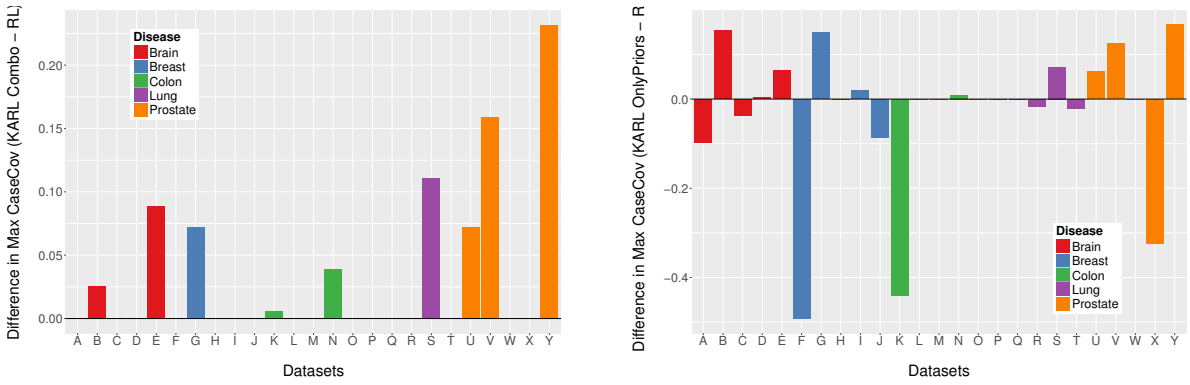

Figure 25: A comparison of maximum positive coverage of tumor examples between KARL and RL. Left: Difference in Max CovPos of tumor examples - KARL Combo vs RL. Right: Difference in Max CovPos of tumor examples - KARL OnlyPriors vs RL.

5.3.7 Difference in Abstentions

Figure 26 represents the difference in the rate of abstaining from predictions between KARL models and baseline RL on the entire datasets. The sub-figure on the left specifically compares KARL Combo vs RL, while the one on the right contrasts KARL OnlyPriors vs RL. As seen, the Combo method reduced the rate of abstentions in the baseline on most datasets, particularly, colon, prostate, and lung cancer sets. Its overall *Wins:Ties:Loses* ratio over the rate of abstentions on the baseline were *19:6:0*. That is, for every datasets, its rate of absten-

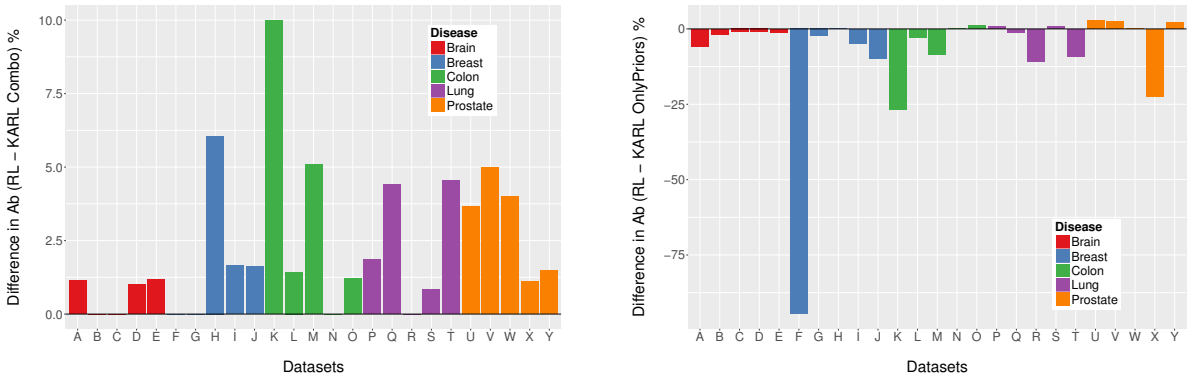

Figure 26: A comparison of the propensity to abstain from predictions between KARL and RL. Left: Difference in rate of Abstentions - RL vs KARL Combo. Right: Difference in rate of Abstentions - RL vs KARL OnlyPriors.

tion was as good as or better than that of baseline. This phenomenon was not surprising as a similar observation was made over coverage, a closely related metric to abstentions. What is more, at a significance level of $\alpha = 0.05$, the reduction in the rate of abstentions by KARL Combo was statistically significant ($p = 0.0001025$ and $p = 0.000143$).

For KARL with only priors search, the rate of abstentions was quite high, especially among the breast and brain cancer sets. Its *Wins:Ties:Loses* ratio over the baseline was *6:3:16*. The increase in abstentions over the baseline was statistically significant ($p = 0.007043$) using Wilcoxon signed-rank test at a significance level of $\alpha = 0.05$.

The apparent increase in abstentions here was largely due to the reason that, this method relies on the source of the background knowledge, and its relationship with the variables of the target dataset. If the overlap between source and target variables are low it is very likely that the target model would suffer from low coverage and high abstentions.

5.3.8 Knowledge Augmented Discovery of robust patterns

Tables 22 to 24 display snippets of robust rule patterns discovered with the knowledge augmented rule-learning framework on brain, lung, and prostate cancer data sets. For brevity, the rest have been provided as additional material (see appendix D). Each table

presents nuggets of general rule patterns that KARL discovered across several datasets within the same disease type. The name of the datasets where the patterns were discovered are presented in bold face font. Within the brain cancer set, for instance, the rule patterns `IF(VEGFA=Low)THEN(Class=Control)` and `IF(VEGFA=High)THEN(Class=Case)` were consistent across KARL models induced from datasets **GEO16011**, **GEO1993**, **GEO4290**, **GEO4271**, and **GEO4412**. In addition, each discovered pattern is numbered and annotated with their provenance; *Pr* denote a prior rule that was retained in the final rule model, while *Nr* signify new rule. Note that some rules are annotated with both *Pr* and *Nr*. Such annotations means that the rule was a retained prior rule in one model, and a new one in another.

Table 22: A snippet of robust rule patterns that were discovered with KARL on brain cancer datasets. **Pr** := rule pattern emanates from prior rules, **Nr** := discovered new rule, **XX** := nomenclature for family of genes

GEO16011, GEO1993, GEO4290, GEO4412
1. `Pr,Nr. IF (COLXXX=Low)  THEN (Class=Control)`
2. `Pr,Nr. IF (COLXXX=High) THEN (Class=Case)`
GEO16011, GEO1993, GEO4290, GEO4271, GEO4412
1. `Pr. IF (VEGFA=Low)  THEN (Class=Control)`
2. `Pr. IF (VEGFA=High) THEN (Class=Case)`
3. `Pr. IF (LDHA=Low)  THEN (Class=Control)`
4. `Nr. IF (LDHA=High)  THEN (Class=Case)`
GEO16011, GEO1993, GEO4290, GEO4271
1. `Pr. IF (IGFBPX=Low)  THEN (Class=Control)`
2. `Pr. IF (IGFBPX=High) THEN (Class=Case)`
GEO16011, GEO1993, GEO4412
1. `Pr. IF (SERPINXX=Low)  THEN (Class=Control)`
2. `Nr. IF (SERPINXX=High) THEN (Class=Case)`

These patterns have some unique features. First, the polarity (i.e., whether the expres-

Table 23: A snippet of robust rule patterns that were discovered with KARL on lung cancer datasets. **Pr** := rule pattern emanates from prior rules, **Nr** := discovered new rule, **XX** := nomenclature for family of genes

GEO10072			
1. Pr,Nr.	IF (EDNRB=High)	THEN	(Class=Control)
2. Nr.	IF (EDNRB=Low)	THEN	(Class=Case)
3. Pr.	IF (PECAM1=High)	THEN	(Class=Control)
4. Pr .	IF (PECAM1=Low)	THEN	(Class=Case)

GEO18842, GEO19188			
1. Pr.	IF (CENPE=Low)	THEN	(Class=Control)
2. Pr.	IF (CENPE=High)	THEN	(Class=Case)
3. Pr,Nr.	IF (PLK4=Low)	THEN	(Class=Control)
4. Pr	IF (PLK4=High)	THEN	(Class=Case)
5. Pr,Nr.	IF (AQPX=High)	THEN	(Class=Control)
6. Pr,Nr.	IF (AQPX=Low)	THEN	(Class=Case)

GEO19804			
1. Pr.	IF (AGER=High)	THEN	(Class=Control)
2. Pr.	IF (AGER=Low)	THEN	(Class=Case)
3. Pr.	IF (CDH3=Low)	THEN	(Class=Control)
4. Pr.	IF (CDH3=High)	THEN	(Class=Case)

GEO19804, GEO18842			
1. Nr.	IF (CCNB1=Low)	THEN	(Class=Control)
2. Pr.	IF (CCNB1=High)	THEN	(Class=Case)

sion intensity of the gene is *High* or *Low*) of each model variable was consistent across all models within which it occurred. For the example pattern above, the assertion that if the variable, `VEGFA`, is highly regulated, predict `Case` held same for all models on **GEO16011**,

Table 24: A snippet of robust rule patterns that were discovered with KARL on prostate cancer datasets. **Pr** := rule pattern emanates from prior rules, **Nr** := discovered new rule

GEO17951, GEO4660, GEO82188
1. `Pr,Nr. IF (HPN = Low)  THEN (Class = Control)`
2. `Pr,Nr. IF (HPN = High) THEN (Class = Case)`
GEO17951, GEO82188
1. `Pr. IF (TRPM4 = Low)  THEN (Class = Control)`
2. `Pr. IF (TRPM4 = High) THEN (Class = Case)`
GEO32448, GEO4660
1. `Pr,Nr. IF (CYP3A5 = High) THEN (Class = Control)`
2. `Pr,Nr. IF (CYP3A5 = Low)  THEN (Class = Case)`
GEO32448, GEO82188
1. `Pr. IF (ID4 = High) THEN (Class = Control)`
2. `Pr. IF (ID4 = Low)  THEN (Class = Case)`

GEO1993, **GEO4290**, **GEO4271**, and **GEO4412**. Second, the patterns display closure on the variables as regards rule structure. That is, when the polarity/direction of a variable value infers one class, its opposite direction concludes the opposite class value as well (see example above). Note that the target class for all datasets were binary (i.e., "Case" or "Control"), and almost all variables were dichotomized into two values (i.e., "Low" or "High") by **EBD**. Third, for some models the closure and complement of a rule was provided by a complimentary source. `LDHA`, for instance, was involved in two complimentary rules (see table 22). While one was new (`Nr. IF (LDHA=High) THEN (Class=Case)`), the other was a retained prior rule (`Pr. IF (LDHA=Low) THEN (Class=Control)`). Last, most of these robust patterns contains variables that belong to gene families (see example of collagen, `COLXXX`, in table 22). The uniqueness about such families was that their polarity was almost consistent wherever they occurred. An example can be found in table 25, snippets

from brain cancer models.

Table 25: Example of unique rules from gene families

Pattern	**Source**
`IF (COL4A2=Low) THEN (Class=Control)`	*GEO1993*
`IF (COL1A1=Low) THEN (Class=Control)`	*GEO4412*
`IF (COL6A1=Low) THEN (Class=Control)`	*GEO16011*

The unique attributes of these robust patterns as elucidated above can be particularly useful for screening, diagnosis, and prognosis of some types of cancer. Though they require further and an in-depth verification studies from domain experts, information contained in majority of then can be verified from literature. Most of the collagen family of genes discovered from the brain cancer (see table 22), for instance, have been implicated in glioblastoma tumorignesis; diffuse invasion of tumor cells into brain tissue typifies the advancement of tumor growth in some type of brain cancer, like glioblastoma [184, 185]. Senner et al. [184] found that the expression of collagen XVI was upregulated in glioblastomas and it promotes tumor cell adhesion. Meanwhile, studies done by Bauer et al. [185] also found out that the inhibition of collagen XVI expression reduces glioma cell invasiveness. This information from the literature suggests that, collagen XVI can be used as viable biomarker to classify brain cancer examples. That is, when collagen XVI is upregulated, predict "Tumor", otherwise predict "Normal". This general, but vital, knowledge was discovered by KARL in `IF (COLXXX=Low) THEN (Class=Control)` and `IF (COLXXX=High) THEN (Class=Case)` (see table 22). Note that the discovery of this knowledge was augmented by information contained in the lookup table on genes associated with cell invasion, on of three major hallmarks of cancer that were considered for this work.

Another example of biomedical significance of knowledge augmented rule pattern discovery, worth discussing, are nuggets of information about the biomarker `AGER`, which was discovered among the lung cancer models (see table 23). `AGER`, also known as `RAGE`, is a member of the immunoglobulin superfamily, and a multifunctional receptor with multiple ligands that have been found to play leading roles in diseases like arthritis, diabetes, and

Alzeimer's [186,187]. Evidence from recent studies indicate that this receptor likely plays an important role in cancer, particularly, its ability to lead cancer cell proliferation, invasion, and survival [186,188,189]. `AGER`, as well as several of its isoforms, is highly expressed in normal lung. However, and unlike other cancers, it is characterized by low expressions in human lung carcinomas [190,191]. Thus, a down-regulated `AGER` would most likely be associated with a late stage of cancer, and therefore suggests it may function as a tumor suppressor for lung cancer. What is more, the general hypothesis, as elicited from literature above, that a highly expressed `AGER` implies "Normal", while the converse is true, was dully captured by KARL; that is `IF (AGER=High) THEN (Class=Control)` and `IF (AGER=Low) THEN (Class=Case)` (see table 23. This is another of several examples that epitomizes KARL's utility as a knowledge augmented classification rule learner.

Last, let us consider `HPN` (*Hepsin*), another biomarker and a variable that was discovered from the prostate cancer models (see table 24). It contains a transmembrane serine protease, which may be in several cellular functions such as cell morphology and blood coagulation [192]. Several studies have identified hepsin as one of the most upregulated genes in prostate cancer [192–194]. Observe that it was predominant and pervasive among most of the prostate cancer models. Similarly, the general notion in literature as regards correlation of its state of expression to prostate cancer progression was transferred into the KARL model. Thus, `IF (HPN = Low) THEN (Class = Control)` and `IF (HPN = High) THEN (Class = Case)`. Combining the evidence revealed from the examples above with other biologically unconfirmed/unverified patterns discovered by KARL turn it into a potent tool for cancer diagnosis and screening.

5.3.9 Results Summary - KARL

Figure 27 summarizes the average classification performance of the two variants of KARL in comparison with baseline RL per disease type. KARL Combo performs better than the baseline, including its specialized version, OnlyPriors, in all five type of cancer. Figure 28, on the other hand, presents a summary of the average positive coverage of "Tumor" and 'Normal" training examples by both version of KARL and RL within each disease type.

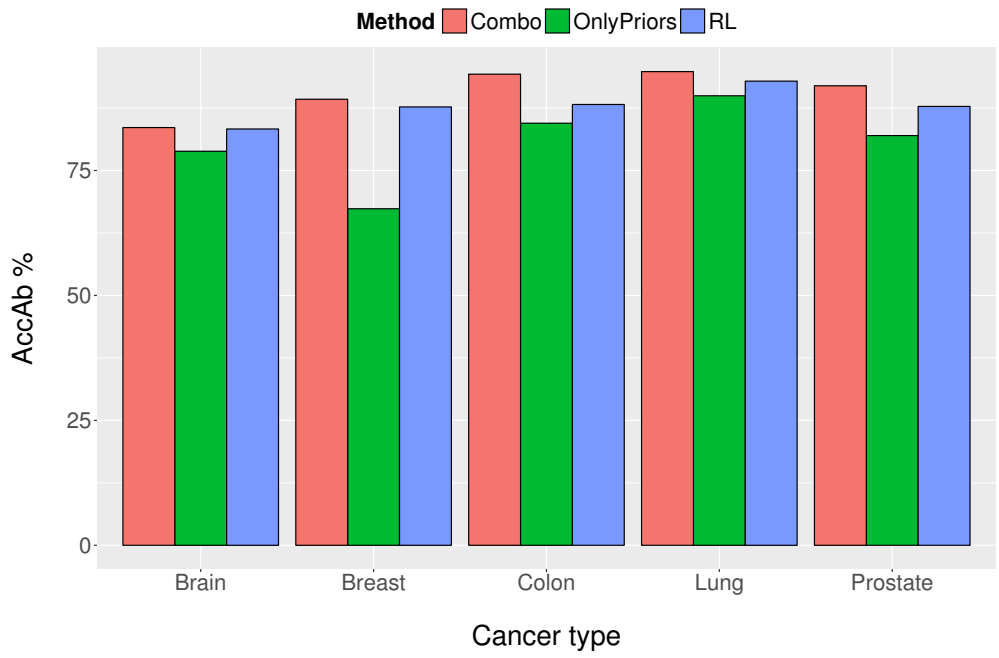

Figure 27: A summary of classification performance of KARL (Combo and OnlyPriors) with baseline RL per cancer type. Performance was measured by accuracy, where abstentions were considered as errors (AccAb)

Similarly, KARL Combo outperforms RL, including its variant, OnlyPriors, on all disease sets. Using these figures, coupled with the analysis in section 5.3.8 we can generalize the characteristics and utility of the KARL framework as follows:

1. KARL Combo, in general, is more *complete* than baseline RL and KARL OnlyPriors.
2. KARL Combo, in general, is more *consistent* than baseline RL and KARL OnlyPriors.
3. KARL OnlyPriors is more *consistent* than baseline RL within the subset of data where it covers as much examples as RL.
4. Rule models induced by KARL provides nuggets of robust patterns, which contains vital domain knowledge.

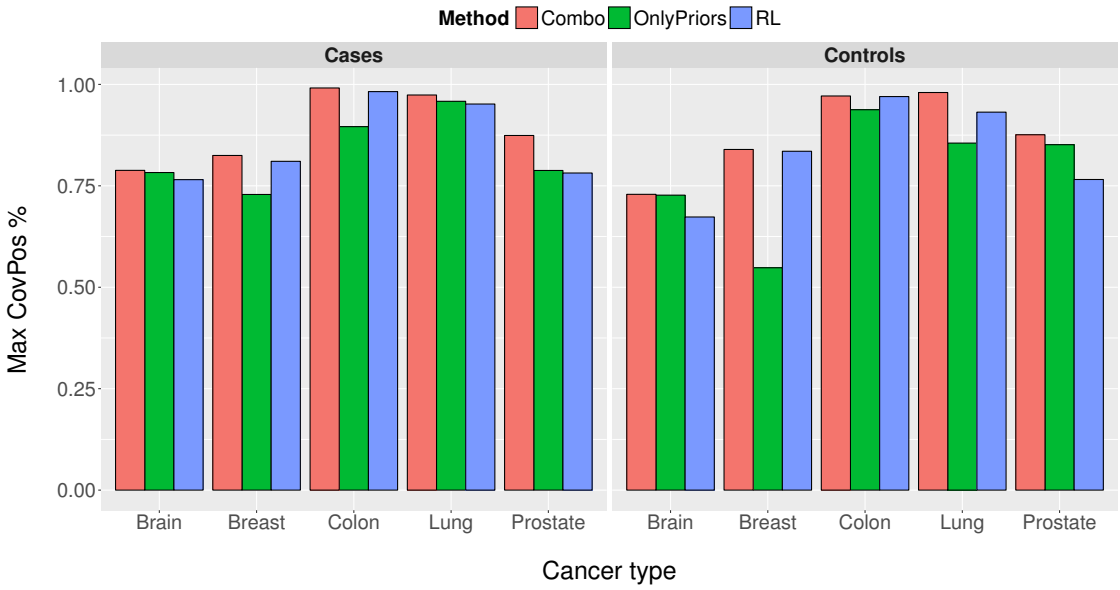

Figure 28: A summary of maximum positive coverage of KARL (Combo and OnlyPriors) with baseline RL per cancer type

5.4 MS-TRL

5.4.1 Classification performance - MS-TRL

Table 26 (and tables 67 and 68 in appendix E) constitute results for classification performance on the brain cancer datasets, using MS-TRL Combo with two, three, and four sources, respectively. Further, table 27 (and tables 69 and 70, also in appendix E) represent predictive performance results for MS-TRL Combo on the same number of sources, but on the *mixed* disease set. Like TRL, each table displays the name of source and target datasets, as well as the predictive performance in sensitivity (SN), specificity (SP), balanced accuracy, accuracies where abstentions are considered errors (AccAb), and the rate of abstaining from making a prediction (Ab). The unit for each performance metric is percent (%).

In general, MS-TRL Combo with two or more sources, more often than not, improves baseline RL performance. Unsurprisingly, though, the performance hinged on that of the

Table 26: Classification performance of MS-TRL Combo, using four (all) source datasets within the brain cancer set

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**
all_sources	GEO16011	96.855	12.500	54.678	89.143
all_sources	GEO1993	92.308	68.421	80.364	84.483
all_sources	GEO4271	98.684	20.833	59.759	80.000
all_sources	GEO4290	95.062	31.579	63.320	83.000
all_sources	GEO4412	98.305	53.846	76.076	84.706

separate TRL (i.e., MS-TRL with a single source) of the sources to the target. If the target performs poorly on its own, multi-source rule transfer would most likely lead to positive transfer. Observe all transfers to GEO16011, GEO4271, GEO4412, and GEO4290 from tables 26, 67 and 68. For GEO1993, which recorded the best performance among the brain cancer cohort, MS-TRL maintained its performance on 4/6 combination of two sources. The performance, however, decreased marginally when the number of sources were increased to three and four.

Table 27: Classification performance of MS-TRL Combo, using four (all) source datasets within the mixed cancer set

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**
all_sources	GEO4412	96.610	50.000	73.305	82.353
all_sources	GEO6956	95.652	55.000	75.326	86.517
all_sources	GEO7670	94.872	81.481	88.177	89.394
all_sources	GEO7904	97.674	31.579	64.627	77.419
all_sources	GEO9348	100.000	66.667	83.333	95.122

For the *mixed* cancer set, where heterogeneity was quite high, the performance of multi-source transfer learning, in general, was low. The observations discussed on the brain cancer set, however, held same, albeit marginally. Applying TRL on GEO6956 and GEO7904 to GEO9348 maintained the BACC of the target (i.e., 91.67), while it increased the AccAb from 95.12 to 97.56. When these two sources were combined for MS-TRL, they still maintained

both the BACC (i.e., for baseline RL and TRL) and AccAb (for TRL). The performance dropped, however, when poorly performing sources, like GEO4412, were added. This was a general trend throughout the MS-TRL experiments. Thus, increasing the number of sources in MS-TRL does not necessarily lead to positive transfer, but finding the right combination of sources is the key to improvement on performance. Future methods can employ other intelligent methods, like genetic algorithms, to identify the right blend and number, if available, of sources for MS-TRL.

Table 28: Classification performance of MS-TRL OnlyPriors, using four (all) source datasets within the brain cancer set

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**	**Ab**
all_sources	GEO16011	92.949	57.143	75.046	87.429	2.857
all_sources	GEO1993	94.737	57.895	76.316	81.034	1.724
all_sources	GEO4271	96.000	43.478	69.739	82.000	2.000
all_sources	GEO4290	88.462	52.632	70.547	79.000	3.000
all_sources	GEO4412	92.857	52.000	72.429	76.471	4.706

Similarly, table 28 (and tables 71 and 72 see appendix E) as well as table 29 (and tables 73 and 74 in appendix E) represent the predictive performances of MS-TRL experiments using the "OnlyPriors" approach on the brain and mixed cancer set respectively. In skewed datasets like the brain cancer, the balanced accuracies increased with increasing number of sources. The general trend as expounded above prevailed here too. That is, with the "right" mix of sources MS-TRL would most likely lead to positive transfer even within subsets of the training data (see the BACC column for tables 28, 71 and 72). Meanwhile, MS-TRL OnlyPriors on the mixed or heterogeneous data sets can be very costly. In general, the abstentions were relatively high, while the classification performances were quite lower.

The general trend of performance of both versions of MS-TRL were pervasive across all the disease sets. The next section presents an average performance per number of sources per disease set.

Table 29: Classification performance of MS-TRL OnlyPriors, using four (all) source datasets within the mixed cancer set

Source	Target	SN	SP	BACC	AccAb	Ab
all_sources	GEO4412	91.379	50.000	70.690	76.471	3.529
all_sources	GEO6956	90.625	60.000	75.312	78.652	5.618
all_sources	GEO7670	94.872	76.000	85.436	84.848	3.030
all_sources	GEO7904	93.023	31.250	62.137	72.581	4.839
all_sources	GEO9348	100.000	50.000	75.000	92.683	0.000

5.4.2 RL vs TRL vs MS-TRL - Classification

Tables 30 and 75 presents average balanced accuracy (BACC) and accuracy (abstentions considered as errors), respectively, per number of sources for MS-TRL Combo experiments. Note that when there is no source (i.e., **None**) RL is implied, while a single source denotes TRL. In addition, observe that there is an additional disease name termed "Mix", which involved a mix of datasets that were randomly selected from each of the five disease sets.

For each target dataset within a disease set, the number of RL, TRL, MS-TRL_2 (MS-TRL with two sources), MS-TRL_3 (MS-TRL with three sources), and MS-TRL_4 (MS-TRL with four sources) experiments were, respectively, 1, 4, 6, 4, 1. As discussed in section 5.4 there were variation in both the BACC and AccAb of the TRL as regards positive/negative transfer. Averaging them out here, for the sake of brevity, may mask some of the trend. Nonetheless, some of the general trend were still preserved.

MS-TRL on the *mixed* set resulted in negative transfer when the number of sources were increased (see the "Mix" block from table 30). This could be attributed to an increase in noise, as the diversity and heterogeneity of the sources increased. For most of the skewed datasets within the brain (e.g., GEO16011, GEO4271, GEO4290, GEO4412), breast(e.g., GEO10780), and colon (e.g., GEO9348, GEO24514) sets there was general increase in positive transfer, due to BACC, when increase sources. The same trend was observed for datasets with relatively high abstentions, like some datasets from the lung (e.g., GEO18842,

Table 30: The average BACC (%) per number of sources for MS-TRL-Combo

		Number of sources				
Disease	**Target**	**None**	**One**	**Two**	**Three**	**Four**
Brain	GEO16011	49.367	51.399	53.013	54.133	54.678
Brain	GEO1993	82.996	82.642	81.883	80.685	80.364
Brain	GEO4271	54.167	58.333	60.326	60.801	59.759
Brain	GEO4290	55.263	61.923	64.609	64.556	63.320
Brain	GEO4412	74.337	74.438	77.586	78.113	76.076
Breast	GEO10780	71.250	72.370	73.170	73.650	73.810
Breast	GEO15852	84.884	85.466	85.078	84.884	86.047
Breast	GEO29431	100.000	91.551	89.969	90.162	90.741
Breast	GEO42568	90.625	89.960	87.345	85.549	84.813
Breast	GEO7904	70.113	69.601	68.229	67.827	67.564
Colon	GEO10715	60.417	69.357	69.820	70.000	71.842
Colon	GEO20916	98.529	98.540	98.543	97.835	97.141
Colon	GEO23878	100.000	97.298	97.168	97.150	98.571
Colon	GEO24514	90.000	94.167	94.755	95.834	96.667
Colon	GEO9348	91.667	93.750	99.306	100.000	100.000
Lung	GEO10072	97.917	96.923	95.748	95.408	95.918
Lung	GEO18842	95.238	97.784	98.152	99.167	100.000
Lung	GEO19188	94.505	95.275	95.018	94.890	94.505
Lung	GEO19804	94.958	94.993	94.445	94.584	95.000
Lung	GEO7670	91.186	93.875	94.041	94.160	93.732
Mix	GEO4412	74.337	74.833	73.546	72.659	73.305
Mix	GEO6956	76.765	75.664	75.791	75.678	75.326
Mix	GEO7904	91.186	88.268	89.625	89.886	88.177
Mix	GEO7670	70.113	67.335	64.973	63.755	64.627
Mix	GEO9348	91.667	89.584	86.806	83.333	83.333
Prostate	GEO17951	86.383	87.079	86.809	86.466	86.136
Prostate	GEO32448	93.611	91.513	92.276	93.117	93.750
Prostate	GEO46602	94.444	90.377	90.062	89.831	91.468
Prostate	GEO6956	76.765	73.813	74.197	71.033	70.326
Prostate	GEO82188	89.346	90.914	90.294	89.686	89.686

Table 31: Pairwise t-test of number of sources (MS-TRL Combo) by BACC

	RL	**TRL**	**MS-TRL_2**	**MS-TRL_3**
TRL	0.7023			
MS-TRL_2	0.6609	0.7109		
MS-TRL_3	0.9186	0.7626	0.2011	
MS-TRL_4	0.8652	0.8675	0.4865	0.7204

Table 32: Pairwise t-test of number of sources (MS-TRL Combo) by AccAb

	RL	**TRL**	**MS-TRL_2**	**MS-TRL_3**
TRL	0.0001217			
MS-TRL_2	0.0002222	0.1743000		
MS-TRL_3	0.0022620	0.6994000	0.3000000	
MS-TRL_4	0.0016820	0.3879000	0.9655000	0.2727000

GEO7670) and colon (e.g., GEO24514) sets. These could be the effect of combining information from multiple related models to improve *completeness*. For baseline RL models that performed well, transfer hardly improved performance. That is, negative transfer was more likely to occur when the number of sources were increased. Examples can be seen from the brain (e.g., GEO1993), breast (e.g., GEO42568, GEO7904), colon (GEO23878), lung (e.g., GEO10072), and prostate (e.g., GEO6956) cancer sets. For the rest, gain/loss of BACC due to transfer were very minimal (see table 30).

With AccAb as performance metric most of the trends expounded above were maintained. The general performance improvement over baseline RL, however, was more significant. Tables 31 and 32 present a pairwise t-test (with significance level of $\alpha = 0.05$) between baseline RL and MS-TRL with increasing number of sources, while using BACC and AccAb as performance metric, respectively.

As recounted above, there was no significant change in BACC, on the average. With AccAb, however, there were significance differences between baseline RL and transfer with

Table 33: The average BACC (%) per number of sources for MS-TRL-OnlyPriors. For number of sources, None ≡ RL, and One ≡ TRL

		Number of sources				
Disease	**Target**	**None**	**One**	**Two**	**Three**	**Four**
Brain	GEO16011	49.367	63.610	70.320	72.705	75.046
Brain	GEO1993	82.996	71.011	74.248	73.881	76.316
Brain	GEO4271	54.167	72.283	72.800	68.223	69.739
Brain	GEO4290	55.263	67.047	69.074	70.521	70.547
Brain	GEO4412	74.337	68.812	73.141	72.022	72.429
Breast	GEO10780	71.250	70.622	79.046	80.164	83.671
Breast	GEO15852	84.884	81.860	88.164	89.014	88.088
Breast	GEO29431	100.000	83.007	86.831	87.895	87.500
Breast	GEO42568	90.625	86.749	87.264	86.404	84.813
Breast	GEO7904	70.113	53.177	60.509	61.306	62.798
Colon	GEO10715	60.417	71.075	74.172	75.980	82.105
Colon	GEO20916	98.529	91.510	93.795	96.016	97.096
Colon	GEO23878	100.000	88.296	93.078	95.882	97.143
Colon	GEO24514	90.000	78.563	92.042	95.932	96.667
Colon	GEO9348	91.667	89.368	95.139	96.875	100.000
Lung	GEO10072	97.917	92.719	94.391	94.850	94.898
Lung	GEO18842	95.238	96.737	98.651	97.707	96.591
Lung	GEO19188	94.505	92.940	93.549	93.868	94.505
Lung	GEO19804	94.958	89.053	91.111	93.071	94.096
Lung	GEO7670	91.186	92.997	94.286	94.748	93.732
Mix	GEO4412	74.337	66.010	70.965	72.308	70.690
Mix	GEO6956	76.765	71.408	73.507	73.478	75.312
Mix	GEO7904	91.186	85.635	88.886	88.819	85.436
Mix	GEO7670	70.113	61.924	63.439	63.396	62.137
Mix	GEO9348	91.667	84.362	77.427	76.042	75.000
Prostate	GEO17951	86.383	80.000	81.804	82.724	85.075
Prostate	GEO32448	93.611	79.966	81.286	83.628	83.487
Prostate	GEO46602	94.444	88.294	88.683	89.965	91.468
Prostate	GEO6956	76.765	70.450	68.625	69.216	69.697
Prostate	GEO82188	89.346	81.691	85.780	86.810	86.221

Table 34: Pairwise t-test of number of sources (MS-TRL OnlyPriors) by BACC

	RL	**TRL**	**MS-TRL_2**	**MS-TRL_3**
TRL	0.0142359833			
MS-TRL_2	0.5393349715	0.000060877		
MS-TRL_3	0.860467784	0.0001042242	0.0189138947	
MS-TRL_4	0.8490666542	0.0001192489	0.0114259531	0.0779507308

at least one source. Though there was performance increase between TRL and its multiple sources variants, the changes were not significant (see table 32).

Unlike tables 30 and 75, tables 33 and 76 show the average balanced accuracy (BACC) and accuracy (abstentions considered as errors), respectively, per number of sources for experiments involving MS-TRL, with prior rules only search. The general pattern as observed from both tables were similar. The prior rules had "little knowledge" about the target, so their initial predictive performance dropped drastically over the baseline. The predictive performance increased significantly, however, as it pooled information from more sources.

The trend is even more striking when abstentions are considered as errors (see table 76). Obviously, restricting search with only priors rules on MS-TRL will more likely lead to increased abstentions; prior rules are more likely to cover limited space within the target domain. Table 34 illustrates the trend with a pairwise t-test of BACC at a significant level of $\alpha = 0.05$ between RL and the multi-source transfer methods with search with only prior rules.

From the table, the significant change in BACC between RL and TRL is due to the drop in performance of the latter. Note, however, that there was no significance difference between RL and the MS-TRL with at least two sources. Meanwhile, there was a stark significance difference between TRL and its other variants. This general pattern as observed in tables 33, 34 and 76 could be particularly useful for cases where new information is not available to augment or confirm prior knowledge for transfer learning. Here, information (classification rules) from multiple related models could be combined to make decisions within a domain.

5.4.3 RL vs TRL vs MS-TRL - Abstentions

Table 35: The average rate (%) of abstentions per number of sources for MS-TRL-Combo

		Number of sources				
Disease	**Target**	**None**	**One**	**Two**	**Three**	**Four**
Brain	GEO16011	1.143	0.686	0.191	0.000	0.000
Brain	GEO1993	0.000	0.000	0.000	0.000	0.000
Brain	GEO4271	0.000	0.000	0.000	0.000	0.000
Brain	GEO4290	1.000	0.250	0.250	0.000	0.000
Brain	GEO4412	1.176	0.000	0.000	0.000	0.000
Breast	GEO10780	1.081	0.676	0.361	0.135	0.000
Breast	GEO15852	0.000	0.000	0.000	0.000	0.000
Breast	GEO29431	6.061	0.379	0.000	0.000	0.000
Breast	GEO42568	1.653	0.207	0.000	0.000	0.000
Breast	GEO7904	1.613	0.403	0.538	0.807	0.000
Colon	GEO10715	13.330	7.450	4.444	3.330	1.429
Colon	GEO20916	1.429	0.715	0.238	0.000	0.000
Colon	GEO23878	5.085	1.695	0.565	0.339	0.000
Colon	GEO24514	0.000	0.000	0.000	0.000	0.000
Colon	GEO9348	2.439	0.610	0.000	0.000	0.000
Lung	GEO10072	1.869	0.468	0.312	0.468	0.935
Lung	GEO18842	4.396	0.274	0.183	0.000	0.000
Lung	GEO19188	0.000	0.000	0.000	0.000	0.000
Lung	GEO19804	0.833	0.208	0.000	0.000	0.000
Lung	GEO7670	4.545	0.000	0.000	0.379	0.000
Mix	GEO4412	1.176	0.588	0.392	0.294	0.000
Mix	GEO6956	1.124	0.843	0.562	0.281	0.000
Mix	GEO7904	1.613	0.000	0.000	0.000	0.000
Mix	GEO7670	4.545	0.379	0.000	0.000	0.000
Mix	GEO9348	2.439	0.000	0.000	0.000	0.000
Prostate	GEO17951	3.650	1.278	0.365	0.183	0.000
Prostate	GEO32448	5.000	1.250	0.625	0.313	0.000
Prostate	GEO46602	4.000	0.000	0.333	0.000	0.000
Prostate	GEO6956	1.124	0.000	0.000	0.000	0.000
Prostate	GEO82188	2.206	0.184	0.000	0.000	0.000

Table 35 displays the average rate of abstaining from making a prediction between RL and

Table 36: Pairwise t-test on the rate of abstentions by number of sources (MS-TRL Combo)

	RL	**TRL**	**MS-TRL_2**	**MS-TRL_3**
TRL	2.987E-06			
MS-TRL_2	4.636E-06	8.869E-05		
MS-TRL_3	6.153E-06	1.114E-02	4.911E-02	
MS-TRL_4	1.266E-05	1.446E-02	3.687E-02	6.279E-02

the MS-TRL methods using combination search. As expected the rate of abstentions increase with an increasing number of sources. As discussed in previous sections, the main reason was that as the algorithm combines information from multiple related domain models, it covers as much space as possible, thereby increasing the baseline completeness. For domains as diverse as the *mixed* and prostate cancer sets, where baseline abstentions were relatively high, increasing the number of sources resulted in a decrease in abstentions.

Table 36 shows a pairwise t-test of the average rate of abstentions based on the number of sources. At a significant level of $\alpha = 0.05$ the difference in abstentions was statistically significant. In addition, the power of the significance increased with increased in the number of sources.

Table 37: Pairwise t-test on the rate of abstentions by number of sources (MS-TRL OnlyPriors)

	RL	**TRL**	**MS-TRL_2**	**MS-TRL_3**
TRL	8.850E-06			
MS-TRL_2	1.051E-02	1.630E-07		
MS-TRL_3	0.4183179	1.657E-06	1.074E-03	
MS-TRL_4	0.6367931	2.638E-06	1.320E-03	7.172E-03

The general trend of the average rate of abstentions per number of sources is pronounced in MS-TRL with only prior rules, albeit with a caveat (see table 77). As observed from predictive performance, the rate of abstentions spikes significantly with a single source, then reduces as the number of sources increased. Table 37, which presents a pairwise test of

significance of the rate of abstentions by number of sources, while using MS-TRL OnlyPriors, summarizes this general trend.

5.4.4 Discovery of Robust patterns with MS-TRL

Tables 38 to 41 illustrate snippet of robust patterns that were discovered from breast, colon, lung, and the *mixed* set models using MS-TRL. See appendix E.2 for more details on the entire disease sets. Recall from section 4.5.1 that a rule pattern is said to be robust in an MS-TRL model if it was induced from more than 50% of source and target datasets—that is, retained from at least two source models.

For tables 38 to 40, the name of the source and target datasets are presented in red and blue font color, respectively. Note also that each rule has been annotated with their provenance. An annotation of the form, say, `S1,S3,S4,T`, denote that the pattern emanated from sources `S1,S2,S3`, and was retained in the target, `T`. For demonstrating closure, some patterns from less than two sources were included. An example is the rule, `S4,T: 1. IF(HILPDA=High) THEN (Class=CASE)`, from the colon cancer set (see table 39), which closed form, `S1,S4,T 2. IF(HILPDA=Low) THEN (Class=CONTROL)`, has been captured already. Like KARL, MS-TRL could be used to discover nuggets of domain knowledge for future verification.

Similarly, these robust patterns may require further verification studies to support and confirm their discovery. Majority of them, however, can be verified from literature. Some markers from the lung cancer set, for instance, have been mentioned in literature findings. `FAM107A` (*family with sequence similarity 107, member A*) is a protein coding gene that is expressed in a variety of normal tissues, and has been implicated in several types of cancer [195–197]. It is downregulated in primary tumors and cell lines, and has been touted as a candidate tumor suppressor gene [195, 197]. Liu et at [196], reported that a downregulated `FAM107A` was identified in non-small cell lung cancer and primary lung cancers. In addition, they recounted that an over-expression of this gene in non-small cell lung cancer lined minimized activities involving cell proliferation and induced apoptosis.

What is more, `FAM107A` was prevalent in several of the MS-TRL models involving lung

Table 38: Examples of robust rule patterns that were discovered by MS-TRL on the combined breast cancer datasets. **S1, S2, S3, S4:=**: Source datasets and also annotated in red font color, **T:=** Target dataset, which have also been marked in blue font color

S1:GEO15852, S2:GEO42568, S3:GEO10780, S4:GEO29431, T:GEO7904
S2,S4,T: 1. IF(ACSL5=Low) THEN (Class=CASE)
S1,S2,S4,T: 2. IF(ADH1B=High) THEN (Class=CONTROL)
S1,S2,T: 3. IF(ADIPOQ=High) THEN (Class=CONTROL)
S2,S4,T: 4. IF(ACADS=High) THEN (Class=CONTROL)
S1,S2,S4,T: 5. IF(ANGPT1=Low) THEN (Class=CASE)
S2,S4,T: 6. IF(ACAA2=Low) THEN (Class=CASE)
S1,S2,S4,T: 7. IF(ABCA8=High) THEN (Class=CONTROL)
S1:GEO7904, S2:GEO15852, S3:GEO42568, S4:GEO29431, T:GEO10780
S2,S3, T 1. IF(ALDH18A1=Low) THEN (Class=CONTROL)
S1,S2,S3,S4,T: 2. IF(ADH1B=High) THEN (Class=CONTROL)
S2,S3,T: 3. IF(CSK=Low) THEN (Class = CONTROL)
S1:GEO7904, S2:GEO42568, S3:GEO10780, S4:GEO29431, T:GEO15852
S1,S2,T: 1. IF(ADH1C=High) THEN (Class=CONTROL)
S2,S4,T: 2. IF(ACACB=Low) THEN (Class=CASE)
S2,S4,T: 3. IF(ACACB=High) THEN (Class=CONTROL)
S1:GEO7904, S2:GEO15852, S3:GEO42568, S4:GEO10780, T:GEO29431
S2,S3,T: 1. IF(COL11A1=High) THEN (Class=CASE)
S2,S3,T: 2. IF(ECT2=High) THEN (Class=CASE)
S1,S2,T: 3. IF(ADH1B=High) THEN (Class=CONTROL)
S2,S3,T: 4. IF(ASPA=High) THEN (Class = CONTROL)
S1:GEO7904, S2:GEO15852, S3:GEO10780, S4:GEO29431, T:GEO42568
S1,S2,T: 1. IF(COMP=High) THEN (Class=CASE)
S2,S4,T: 2. IF(GOS2=High) THEN (Class=CONTROL)
S1,S2,T: 3. IF(FN1=High) THEN (Class=CASE)

Table 39: Examples of robust rule patterns that were discovered by MS-TRL on the combined colon cancer datasets. **S1, S2, S3, S4:=**: Source datasets and also annotated in red font color, **T:=** Target dataset, which have also been marked in blue font color

S1:GEO24514, S2:GEO10715, S3:GEO20916, S4:GEO23878, T:GEO9348
S1,S4,T: 1. IF(HILPDA=Low) THEN (Class=CONTROL)
S3,S4,T: 2. IF(CDH3=High) THEN (Class=CASE)
S3,T: 3. IF(CDH3=Low) THEN (Class=CONTROL)
S1:GEO24514, S2:GEO9348, S3:GEO20916, S4:GEO23878, T:GEO10715
S1,S3,T: 1. IF(CNNM2=High) THEN (Class=CONTROL)
S1:GEO24514, S2:GEO9348, S3:GEO10715, S4:GEO23878, T:GEO20916
S4,T: 1. IF(HILPDA=High) THEN (Class=CASE)
S1,S4,T 2. IF(HILPDA=Low) THEN (Class=CONTROL)
S1:GEO24514, S2:GEO9348, S3:GEO10715, S4:GEO20916, T:GEO23878
S2,S4,T: 1. IF(CA7=High) THEN (Class=CONTROL)
S2,T: 2. IF(CA7=Low) THEN (Class=CASE)
S1:GEO9348, S2:GEO10715, S3:GEO20916, S4:GEO23878, T:GEO24514
S3,S4,T: 1. IF(CDH3=High) THEN (Class=CASE)
S3,S4,T: 2. IF(CDH3=Low) THEN (Class=CONTROL)

cancer. Meanwhile, the notion in literature that it is downregulated in primary lung cancer was confirmed by its general rule pattern (see table 40). Similarly, rules involving variables (e.g., `FAM189A1`, and `FAM189A2`) of its family formed a closure. Other prevalent variables that were involved in the lung cancer set models were `AGER` and `EDNRB`. `AGER`, a member of the immunoglobulin superfamily, and a multifunctional receptor with multiple ligands is known to be associated with stages of several disease, particularly the late stage of lung cancer (see section 5.3.8). Like KARL, the MS-TRL models induced patterns that confirmed literature reports that it may be a potential tumor suppressor [190]. Lastly, `EDNRB`, encodes a G protein-coupled receptor that activates a phosphatidylinositol-calcium second messenger system (RefSeq Accession: NM_000115). It has been implicated in hypoxia and Hirschsprung's disease [198, 199]. One of its important *paralogs* is `NMBR` (*Neuromedin B Receptor*) has also

Table 40: Examples of robust rule patterns that were discovered by MS-TRL on combined lung cancer datasets. **S1, S2, S3, S4:=**: Source datasets and also annotated in red font color, **T:=** Target dataset, which have also been marked in blue font color

S1:GEO19804, S2:GEO10072, S3:GEO18842, S4:GEO19188, T:GEO7670
S1,T: 1. IF(AHNAK=High) THEN (Class=CONTROL)
T: 2. IF(AHNAK=Low) THEN (Class=CASE)
S1,S4,T: 3. IF(EMP2=High) THEN (Class=CONTROL)
S1:GEO19804, S2:GEO7670, S3:GEO18842, S4:GEO19188, T:GEO10072
S2,S4,T: 1. IF(CLIC5=Low) THEN (Class=CASE)
S1,S4,T: 2. IF(ARHGEF15=High) THEN (Class=CONTROL)
S1,S2,S4,T: 3. IF(FAM107A=Low) THEN (Class=CASE)
S1,S2,T: 4. IF(FAM189A2=High) THEN (Class=CONTROL)
S1,S2,S4,T: 5. IF(AGER=High) THEN (Class=CONTROL)
S1,S4,T: 6. IF(EFNA4=High) THEN (Class=CASE)
S1:GEO19804, S2:GEO10072, S3:GEO7670, S4:GEO19188, T:GEO18842
S4,T: 1. IF(AQP1=Low) THEN (Class=CASE)
T: 2. IF(AQP4=High) THEN (Class=CONTROL)
S1,S2,T: 3. IF(FAM107A=Low) THEN (Class=CASE)
S2,S3,T: 4. IF(EDNRB=Low) THEN (Class=CASE)
S2,T: 5. IF(EDNRB=High) THEN (Class=CONTROL)
S1:GEO19804, S2:GEO10072, S3:GEO7670, S4:GEO18842, T:GEO19188
S2,S3,T: 1. IF(EDNRB=Low) THEN (Class=CASE)
S1:GEO10072, S2:GEO7670, S3:GEO18842, S4:GEO19188, T:GEO19804
S2,T: 1. IF(FAM189A2=High) THEN (Class=CONTROL)
S4,T: 2. IF(FAM107A=Low) THEN (Class=CASE)
T: 3. IF(FAM189A1=Low) THEN (Class=CASE)
S2,T: 4. IF(AGER=Low) THEN (Class=CASE)
S1,T: 5. IF(AGER=High) THEN (Class=CONTROL)
S2,S4,T: 6. IF(FABP4=Low) THEN (Class=CASE)
S4,T: 7. IF(ALDH3B2=High) THEN (Class=CASE)
S1,T: 8. IF(ALDH18A1=Low) THEN (Class=CONTROL)

been implicated in small cell lung cancer [200].

Comparatively, models on the colon cancer set recorded one of the best predictive performances among the frameworks, in general. In addition, most of the colon cancer models were relatively short in size (i.e., more parsimonious), and three of the variables that featured predominantly were `HILPDA`, `CA7`, and `CDH3` (see table 39). `HILPDA` (*Hypoxia Included Lipid Droplet-Associated*) is a biomarker of hypoxia and known to be elevated in various forms of cancer [201–203]. Kim et al. [203] recounted that this variable promotes colorectal cancer progression. Results from their studies revealed that an over-expressed `HILPDA` promoted tumor growth by inhibiting apoptosis. `CA7` (*Carbonic Anhydrase VII*) is a member of the carbonic anhydrase family, which participates in a variety of biological processes (e.g., respiration, calcification, acid-base balance, and bone resorption-RefSeq: NM_001014435), and implicated in the pathogenesis of several human cancers [204]. `CA7` has been reported to be expressed is several normal tissues including colon [205]. Meanwhile, other studies have associated a downregulated `CA7` to colon tumors, and it's been implicated as an important suppressor gene for classifying normal and CRC tissues [206, 207]. Results from Yang et al. [208] indicated that a decreased expression of `CA7` correlated with disease progression and poor prognosis of CRC. Last, `CDH3`, a member of the cadherin superfamily, is involved in several cellular processes such as differentiation, embryonic development, cell polarity, growth and migration [209]. It has been implicated in various human tumors, including CRC; upregulated `CDH3` is associated with malignant CRC [210, 211]. Generally, the pattern of association involving these variables and CRC, as reported in literature, were detected and confirmed by the MS-TRL robust rule pattern set.

Unlike tables 38 to 40, table 41 shows robust rule patterns that were discovered from the *mixed* disease set, to demonstrate the utility of MS-TRL as a potential tool for conducting cross-domain studies, like *pan-cancer studies*. Observe that, here, the annotations for rule provenance, by font color, are slightly different. Red, blue, green, purple, and orange font colors were used to represent brain, breast, colon, lung, and prostate cancer, respectively.

These patterns are potential candidates for domain-independent rules. Similarly, thorough verification studies are required to confirm their integrity. Most of the variables involved, however, have been implicated in diverse forms of cancer. `CASP8` is a member of the

Table 41: Examples of robust rule patterns that were discovered by MS-TRL on combined set of randomly mixed cancer datasets. **S1, S2, S3, S4:=**: Source datasets, **T:=** Target dataset, and color annotations (Brain := Red, Breast := Blue, Colon := Green, Lung := Purple, Prostate := Orange) denote the cancer type.

S1:GEO6956, S2:GEO7904, S3:GEO9348, S4:GEO7670, T:GEO4412
S1,S2,T: 1. IF(CASP8=High) THEN (Class=CASE)
S2,S4,T: 2. IF(ABLIM1=High) THEN (Class=CONTROL)
S3,S4,T: 3. IF(CALU=High) THEN (Class=CASE)
T: 4. IF(COL5A2=Low) THEN (Class=CONTROL)
S2,S4,T: 5. IF(COL5A1=High) THEN (Class=CASE)
S1:GEO7904, S2:GEO9348, S3:GEO7670, S4:GEO4412, T:GEO6956
S1,S4,T: 1. IF(BAG1=Low) THEN (Class=CASE)
S1,S2,T: 2. IF(ACADS=High) THEN (Class=CONTROL)
S1,S4,T: 3. IF(DHX9=Low) THEN (Class=CASE)
S1:GEO6956, S2:GEO7904, S3:GEO9348, S4:GEO4412, T:GEO7670
S1,S4,T: 1. IF(CGRRF1=Low) THEN (Class=CASE)
S1:GEO6956, S2:GEO9348, S3:GEO7670, S4:GEO4412, T:GEO7904
S2,S3,S4,T: 1. IF(ABCG2=Low) THEN (Class=CASE)
S3,S4,T: 2. IF(AKT3=Low) THEN (Class=CASE)
S1,S2,T: 3. IF(ACADS=High) THEN (Class=CONTROL)
S1,T: 4. IF(ACADS=Low)&(ADAMTS2=High) THEN (Class=CASE)
S1,S3,T: 5. IF(ANGPT1=Low) THEN (Class=CASE)
S1,S2,S3,T: 6. IF(ACAA2=Low) THEN (Class=CASE)
S3,S4,T: 7. IF(ABLIM1=High) THEN (Class=CONTROL)

caspase family that are involved in the signaling pathways of cell death (apoptosis, necrosis) and inflammation [212]. It is expressed in almost all kinds of tissue, which suggests that its aberrant form can implicate different type of cancer. `CASP8` has been cited to affect metastasis. Loss of `CASP8`, in general, potentiates metastasis; however other studies have reported that it can promote tumor cell migration and metastasis, under conditions where apoptosis is compromised [213, 214]. Thus its association to various type of cancer.

`BAG1` encodes a multifunctional protein, and like `CASP8`, is associated to survival of the cell—it can block a step in a pathway leading to apoptosis. In addition, it regulates other cellular processes such as proliferation, transcription, proteasome-mediated degradation, and metastasis [215]. It is expressed ubiquitously, and found to be upregulated in several forms of human cancer like breast, colorectal, prostate, lung, esophageal, and squamous cell carcinoma [216–218].

The examples discussed above indicates that MS-TRL can be used to isolate candidate domain-specific (or independent) patterns for further verification. The role and effect of a variable to one type of cancer may not be necessarily same for another, especially for multifunctional genes that might influence multiple cellular processes in diverse ways. This means that the findings from these patterns cannot be blindly generalized. Nonetheless, they could provide immense insight for integrative studies of related gene expression studies.

5.4.5 Relatedness & transfer

The Venn diagram in Figure 29 depicts the distribution of domain-independent variables, as measured with mutual information, among the *mixed* datasets. Recall (see section 4.5.1) that a variable is considered domain-independent if the mutual information, MI, between it and the class variable is $MI \leq 0.1$. The MI for each variable in within a disease set was evaluated, but for brevity, I opted to illustrate the *mixed* set; relatively, it is more interesting since it involves datasets that were drawn from diverse disease type, which can also be classified as unique domains.

The color encoding for each disease representative are red, green, purple, orange, and cyan for breast (GEO7904), colon (GEO9348), lung (GEO7670), prostate (GEO6956), brain

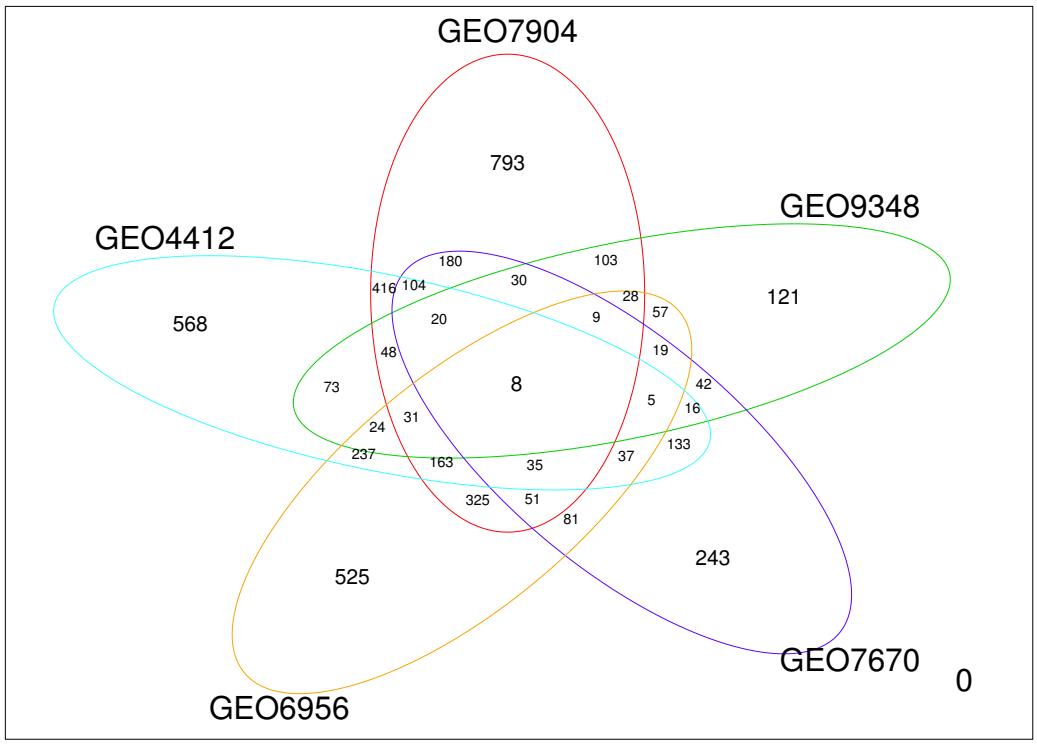

Figure 29: Distribution of domain-independent variables among mixed datasets

(GEO4412), respectively. In all, the total number of domain-independent variables within each disease rep was 2344, 634, 1013, 1635, and 1918 for GEO7904, GEO9348, GEO7670, GEO6956, and GEO4412, in that order. Thus the colon and breast cancer reps recorded the least and most domain-independent variables, respectively.

Table 42 represents the relationship between the direction of transfer (i.e., positive or negative) and the fraction of domain-independent variable within the target that the source (i.e., **FracDI**). Here, positive transfer is deemed as a gain in accuracy over the target due to transfer from the source. Positive transfer is denoted by the indicator value 1, while 0 represent negative transfer.

Generally, there were positive transfers from all sources to GEO9348. This could probably be due to its overall low number of domain-independent variables. The converse of this

Table 42: Relationship between relatedness & positive transfer. **FracDI** = fraction of total domain-independent variables in the target, shared by source. Direction of transfer; *positive* = 1; *negative* = 0

Source	**Target**	**FracDI**	**Transfer**
GEO6956	GEO4412	0.282	1
GEO7670	GEO4412	0.187	0
GEO7904	GEO4412	0.430	1
GEO9348	GEO4412	0.117	0
GEO4412	GEO6956	0.330	0
GEO7670	GEO6956	0.150	0
GEO7904	GEO6956	0.398	1
GEO9348	GEO6956	0.111	0
GEO4412	GEO7670	0.353	1
GEO6956	GEO7670	0.242	0
GEO7904	GEO7670	0.431	1
GEO9348	GEO7670	0.147	1
GEO4412	GEO7904	0.352	1
GEO6956	GEO7904	0.277	0
GEO7670	GEO7904	0.186	1
GEO9348	GEO7904	0.118	0
GEO4412	GEO9348	0.355	1
GEO6956	GEO9348	0.285	1
GEO7670	GEO9348	0.235	1
GEO7904	GEO9348	0.437	1

hypothesis, however, can not be established for GEO7904, which had the most number of variables. The trend that stood out, in general, was that for all possible sources to a particular target, the one with the most **FracDI**, invariably, led to positive transfer. This could be an invaluable indicator for selecting which source(s) to use for transfer, as, in some cases, negative transfer due to just one source can have a costly effect, downstream, on the performance of a multi-source transfer rule learning model.

5.4.6 Results Summary - MS-TRL

From the experimental results discussed thus far, the following conclusions can be summarized about the MS-TRL framework:

1. The MS-TRL framework statistically significantly improves the classification performance of baseline RL. Thus, it is more *consistent* than RL.
2. For most cases, transfer rule learning models with multiple sources are more accurate than transfer with a single source. Increasing, the number of sources, however, does not guarantee improvement in performance. Intelligent approaches for selecting sources that are more likely to lead to positive transfer are required; negative effects from some source(s) can ruin the performance downstream.
3. The MS-TRL framework statistically significantly reduces the rate of abstentions. There is an inverse relationship between the number of sources and rate of abstentions. The more the sources the less it abstains from making a prediction. Multiple sources increases coverage of the domain space, and hence the framework is more *complete* than baseline RL.
4. MS-TRL is sufficient for discovering robust nuggets of rule patterns that are dependent or independent of a domain. Majority of the discovered knowledge can be verified from literature evidence.
5. With the availability of more than one source the one which shares the most domain-independent variables of the target is most likely to lead to positive transfer.

5.5 iTRL

5.5.1 Classification performance - iTRL

Section 5.5.1 presents and discusses results from the iTRL cross-validation experiments. Due to the permutations and ordering of the sources, there were 300 experiments, and results, for each disease set. For brevity, the results involving one dataset (i.e., GEO16011, brain cancer) as target, in all permutation scenarios, was selected to highlight the impact and relevance of ordering as far as sources are concerned as regards incremental transfer rule learning.

Tables 43, 81 and 82 represent cross-validation results of iTRL Combo experiments, while using the brain cancer data, GEO16011, as target. The number of sources for each results set is in the order of 2, 3, and 4 respectively. In addition, note that each table presents the possible permutation of ordering from a given set of sources.

Although, the set of sources were the same for each set of results, there were variations in both BACC and AccAb. The values of BACC ranged from $49.057\% - 55.621\%$, while AccAb was $89.143\% - 90.857\%$ for two and three sources and $87.429\% - 90.857\%$ for four sources. The variations in the accuracies supports the notion that "ordering of sources, indeed, matters for incremental learning via transfer rule learning."

Like MS-TRL, combining sources, which individually improve the baseline accuracies, more likely lead to positive transfer. The magnitude of the transfer, however, depends on the ordering. From tables 13 and 81, GEO4271 and GEO1993, individually, improved baseline accuracy, significantly as compared to the others within the set, and their combination, thereof, as sources also led to positive transfer. Ordering them yielded slightly different accuracies. The best accuracy within the set, however, emanated from a combination of GEO1993 and GEO4290, which had low specificity on its own. It was a better pairing for GEO1993 than the rest, however, due to its sensitivity (highest in the group). Like MS-TRL, combining the sources with highest sensitivity and specificity, invariably, led to positive transfer, getting the ordering right will maximize the accuracy—a potential future study to explore.

Tables 44, 83 and 84, on the other hand, present cross-validation results on the same

Table 43: Classification performance of iTRL Combo, with four sources on GEO16011, brain cancer

Source	Target	SN	SP	BACC	AccAb
GEO4290_GEO4412_GEO1993_GEO4271	GEO16011	95.60	6.25	50.92	87.43
GEO4412_GEO1993_GEO4290_GEO4271	GEO16011	95.60	6.25	50.92	87.43
GEO4412_GEO4290_GEO1993_GEO4271	GEO16011	95.60	12.50	54.05	88.00
GEO4290_GEO1993_GEO4271_GEO4412	GEO16011	96.86	6.25	51.55	88.57
GEO1993_GEO4290_GEO4412_GEO4271	GEO16011	96.86	6.25	51.55	88.57
GEO1993_GEO4412_GEO4290_GEO4271	GEO16011	96.86	6.25	51.55	88.57
GEO4290_GEO1993_GEO4412_GEO4271	GEO16011	97.48	6.25	51.87	89.14
GEO1993_GEO4290_GEO4271_GEO4412	GEO16011	97.48	6.25	51.87	89.14
GEO4271_GEO1993_GEO4412_GEO4290	GEO16011	98.11	0.00	49.06	89.14
GEO1993_GEO4412_GEO4271_GEO4290	GEO16011	98.11	0.00	49.06	89.14
GEO4290_GEO4271_GEO1993_GEO4412	GEO16011	98.11	6.25	52.18	89.71
GEO4412_GEO1993_GEO4271_GEO4290	GEO16011	98.11	6.25	52.18	89.71
GEO1993_GEO4271_GEO4412_GEO4290	GEO16011	98.73	0.00	49.37	89.14
GEO4271_GEO4412_GEO1993_GEO4290	GEO16011	98.11	6.67	52.39	89.71
GEO1993_GEO4271_GEO4290_GEO4412	GEO16011	98.73	6.25	52.49	89.71
GEO4412_GEO4290_GEO4271_GEO1993	GEO16011	98.11	12.50	55.31	90.29
GEO4290_GEO4412_GEO4271_GEO1993	GEO16011	98.11	12.50	55.31	90.29
GEO4412_GEO4271_GEO1993_GEO4290	GEO16011	98.11	12.50	55.31	90.29
GEO4290_GEO4271_GEO4412_GEO1993	GEO16011	98.11	12.50	55.31	90.29
GEO4271_GEO1993_GEO4290_GEO4412	GEO16011	98.11	12.50	55.31	90.29
GEO4271_GEO4290_GEO1993_GEO4412	GEO16011	98.74	6.67	52.70	90.29
GEO4271_GEO4412_GEO4290_GEO1993	GEO16011	98.74	12.50	55.62	90.86
GEO4271_GEO4290_GEO4412_GEO1993	GEO16011	98.74	12.50	55.62	90.86
GEO4412_GEO4271_GEO4290_GEO1993	GEO16011	98.74	12.50	55.62	90.86

Table 44: Classification performance of iTRL OnlyPriors, with four sources on GEO16011, brain cancer

Source	Target	SN	SP	BACC	AccAb
GEO4271_GEO4290_GEO1993_GEO4412	GEO16011	94.29	0.00	47.14	75.43
GEO4412_GEO1993_GEO4290_GEO4271	GEO16011	96.43	0.00	48.21	46.29
GEO4271_GEO4412_GEO4290_GEO1993	GEO16011	96.95	0.00	48.47	72.57
GEO4271_GEO4290_GEO4412_GEO1993	GEO16011	97.06	0.00	48.53	75.43
GEO4412_GEO1993_GEO4271_GEO4290	GEO16011	97.83	0.00	48.91	51.43
GEO4412_GEO4290_GEO1993_GEO4271	GEO16011	97.90	0.00	48.95	53.14
GEO4290_GEO4412_GEO1993_GEO4271	GEO16011	99.03	0.00	49.52	58.29
GEO4290_GEO1993_GEO4271_GEO4412	GEO16011	99.06	0.00	49.53	60.00
GEO4290_GEO1993_GEO4412_GEO4271	GEO16011	99.08	0.00	49.54	61.71
GEO4412_GEO4290_GEO4271_GEO1993	GEO16011	99.18	0.00	49.59	69.14
GEO4412_GEO4271_GEO1993_GEO4290	GEO16011	99.18	0.00	49.59	69.14
GEO4412_GEO4271_GEO4290_GEO1993	GEO16011	99.19	0.00	49.59	69.71
GEO4290_GEO4412_GEO4271_GEO1993	GEO16011	100.00	0.00	50.00	63.43
GEO4290_GEO4271_GEO1993_GEO4412	GEO16011	100.00	0.00	50.00	56.00
GEO4290_GEO4271_GEO4412_GEO1993	GEO16011	100.00	0.00	50.00	57.14
GEO4271_GEO1993_GEO4412_GEO4290	GEO16011	93.99	33.33	63.66	72.57
GEO4271_GEO4412_GEO1993_GEO4290	GEO16011	94.78	33.33	64.06	73.71
GEO1993_GEO4412_GEO4271_GEO4290	GEO16011	98.94	33.33	66.14	54.29
GEO1993_GEO4412_GEO4290_GEO4271	GEO16011	98.97	33.33	66.15	56.00
GEO4271_GEO1993_GEO4290_GEO4412	GEO16011	96.99	37.50	67.25	75.43
GEO1993_GEO4271_GEO4412_GEO4290	GEO16011	96.67	50.00	73.33	51.43
GEO1993_GEO4290_GEO4412_GEO4271	GEO16011	97.90	50.00	73.95	54.29
GEO1993_GEO4290_GEO4271_GEO4412	GEO16011	97.92	50.00	73.96	54.86
GEO1993_GEO4271_GEO4290_GEO4412	GEO16011	96.63	60.00	78.32	50.86

dataset, GEO16011, from the brain cancer set, using iTRL with only prior rules search. The accuracies range about $47\% - 78\%$ for both BACC and AccAb.

As observed throughout results from the other frameworks, rule space search with only prior rules performs very well on subsets of the data examples, which they cover. This pattern was no different from the iTRL. The BACC were much better than the "Combo" version as reported in tables 43, 81 and 82. The AccAb, however, were, comparatively, very low. This was caused by a relatively high rate of abstentions.

Generally, the trend for the rate of abstentions due to iTRL, particularly OnlyPriors search, were different. Unlike MS-TRL OnlyPriors, where the rate of abstentions decreased with increasing number of sources, abstentions withing iTRL (OnlyPriors) increases with the number of iterations; that is, 69.71, 64.31, and 61.76, respectively, for tables 44, 83 and 84. See appendix F.1 for detailed iTRL classification results on the entire datasets.

By combining snippets of rule models from multiple related sources, the MS-TRL would likely increases the coverage space of data examples, therefore reducing the rate of abstentions. The case for iTRL is slightly different as the source models are not lumped, àla MS-TRL style, but rather, merged incrementally—one at a time. That is, when the sources emanate from heterogeneous domains they are more likely to cover diverse spaces, if at all, within the test, and the more it abstains from making predictions. See more examples from appendix F.1, particularly, result on the *mixed* cancer set, where heterogeneity was relatively high.

5.5.2 RL vs TRL vs MS-TRL vs iTRL - classification

The results discussed in sections 5.4.2 and 5.5.1 suggest that, in general, transfer learning improves the baseline classification performance. The more accurate individual sources are the more likely their combination, into a multi-source transfer, leads to positive transfer. It also emerged from the above results that getting the "right mix" and "ordering" of sources, more often than not, led to positive transfer.

Table 45 highlights a comparison of accuracies (including abstentions) from all frameworks given a target dataset. Note that RL, the overarching baseline, can be considered

Table 45: A comparison of best accuracies via all frameworks. **Src** = number of sources

	RL	**TRL**	**MS-TRL**			**iTRL**		
Target	Src=0	Src=1	Src=2	Src=3	Src=4	Src=2	Src=3	Src=4
GEO16011	89.14	90.29	90.86	90.86	89.14	90.86	90.86	90.86
GEO1993	86.21	87.93	87.93	86.21	84.48	89.66	89.66	89.66
GEO4271	78.00	81.00	81.00	81.00	80.00	82.00	82.00	82.00
GEO4290	82.00	85.00	86.00	84.00	83.00	86.00	87.00	85.00
GEO4412	81.18	84.71	87.06	85.88	84.71	85.88	85.88	85.88
GEO10780	86.49	88.11	88.65	88.65	88.11	88.65	88.65	88.11
GEO15852	84.88	86.05	87.21	86.05	86.05	87.21	88.37	88.37
GEO29431	93.94	96.97	96.97	96.97	95.46	95.46	93.94	93.94
GEO42568	95.87	96.69	96.69	95.87	95.04	95.87	95.87	97.52
GEO7904	77.42	79.03	79.03	77.42	77.42	79.03	79.03	79.03
GEO10715	60.00	70.00	70.00	73.33	70.00	73.33	76.67	76.67
GEO20916	97.14	100.00	100.00	100.00	97.14	100.00	100.00	100.00
GEO23878	94.92	100.00	98.31	98.31	98.31	100.00	100.00	98.31
GEO24514	93.88	97.96	97.96	97.96	97.96	97.96	97.96	97.96
GEO9348	95.12	100.00	100.00	100.00	100.00	100.00	100.00	100.00
GEO10072	96.26	98.13	97.20	96.26	95.33	98.13	98.13	97.20
GEO18842	91.21	98.90	100.00	100.00	100.00	98.90	100.00	100.00
GEO19188	94.87	95.51	95.51	95.51	94.87	95.51	95.51	95.51
GEO19804	94.17	95.83	95.00	95.00	95.00	95.83	95.83	95.83
GEO7670	87.88	95.46	95.46	95.46	93.94	96.97	96.97	96.97
GEO4412	81.18	85.88	83.53	83.53	82.35	85.88	88.24	88.24
GEO6956	87.64	88.76	88.76	87.64	86.52	88.76	88.76	88.76
GEO7670	87.88	90.91	92.42	92.42	89.39	92.42	92.42	92.42
GEO7904	77.42	80.65	79.03	77.42	77.42	80.65	80.65	80.65
GEO9348	95.12	97.56	97.56	96.34	95.12	98.78	98.78	98.78
GEO17951	83.21	86.86	87.59	87.59	86.13	88.32	89.78	89.78
GEO32448	88.75	93.75	93.75	93.75	93.75	95.00	95.00	95.00
GEO46602	92.00	98.00	98.00	96.00	94.00	98.00	98.00	98.00
GEO6956	87.64	88.76	91.01	85.39	84.27	89.89	89.89	88.76
GEO82188	87.50	91.91	91.91	89.71	89.71	93.38	93.38	91.91

as learning without any prior knowledge—that is, no source (Src = 0). In addition, TRL can be considered as a variant of both MS-TRL and iTRL, albeit with a single source. Apart from RL, the results from the other frameworks represent the best accuracy among all possible sources. For the brain cancer dataset GEO16011, for instance, as target, the best accuracy from TRL resulted from the source GEO1993. Similarly, {GEO1993, GEO4290} and {GEO1993, GEO4290, GEO4412} were the best sources form MS-TRL with two and three sources, respectively. Meanwhile, the ordering {GEO4290, GEO1993}, {GEO4271, GEO4290, GEO1993}, and {GEO4412, GEO4271, GEO4290, GEO1993}, respectively, were the best source set for iTRL with two, three, and four source (i.e. iterations).

Note that accuracies for MS-TRL with all sources (i.e., four sources) is likely to be inferior to the other transfer variants. This is because there were no alternate sets of four. All four were merged, and as discussed in section 5.4.2, a lowly performing source among the lot might ruin the accuracy of the entire source set—àla "a bad apple spoils the bunch." See appendix F.1 for a detailed list of all "best source(s)", including their accuracies, for each target dataset while using TRL, MS-TRL, and iTRL.

In general, the "best" source from each transfer learning method beats baseline RL as far as accuracy is concerned. Thus, for all things being equal, there exist a source or set of sources, which when used for transfer rule learning—single or multiple—would most likely lead to positive transfer. That said, the notion that "multiple source rule transfer is significantly better than single" might be conditionally dependent on the accuracy of the individual sources themselves. While this condition, in most cases, is necessary for MS-TRL, in iTRL, the "right" ordering of these "good" sources are also required for the above notion to hold.

Table 46 provides a summary of wins, draws, and losses by accuracy of the best TRL versus that of the multiple source variants, while using the same target. The results suggests that, on the same target dataset the best single source rule transfer is more accurate than the multiple rule source (àla MS-TRL) as the number of sources increases. The main reason, as alluded to above, hinged on the accuracies of the individual sources that constitute the source. With incremental learning (àla iTRL), however, there seem to exist paths or ordering of multiple sources, which outperforms the best single source transfer. A potential future

Table 46: Summary of wins, draws, and loses in best accuracy of the MS methods versus TRL, given the same target. MS = Multi-source.

MS Methods	**Wins**	**Draws**	**Losses**
MS-TRL_2	9	16	5
MS-TRL_3	7	9	14
MS-TRL_4	1	7	22
iTRL_2	15	13	2
iTRL_3	17	11	2
iTRL_4	14	13	3

study, which identifies these "sweet spots" or ordering, independently, could be crucially important for incremental transfer learning, particularly when the sources are large in order to avoid fruitless knowledge transfer.

Table 47: Pairwise t-test on the best accuracy per framework

	RL	TRL	MS-TRL_2	MS-TRL_3	MS-TRL_4	iTRL_2	iTRL_3
TRL	4.6E-09						
MS-TRL_2	5.1E-09	4.9E-01					
MS-TRL_3	2.4E-05	5.3E-02	1.5E-02				
MS-TRL_4	1.7E-03	1.5E-06	1.4E-06	3.3E-06			
iTRL_2	9.0E-09	1.6E-03	3.0E-02	1.5E-04	4.5E-08		
iTRL_3	9.5E-08	4.6E-03	2.3E-02	7.3E-05	1.6E-07	7.2E-02	
iTRL_4	2.3E-07	3.0E-02	1.0E-01	1.9E-04	4.8E-07	6.4E-01	1.1E-01

Table 47 presents a paired t-test, at a significance level of $\alpha = 0.05$, of the difference in accuracies by the rule learning frameworks as displayed in table 46. This results confirms that, within a disease set (homogeneous or otherwise), there exit a source or set of sources that when used for transfer learning, statistically significantly outperforms the accuracy of learning without transfer. What is more, the power of the significant difference in accuracies was more pronounced in transfers with at most two sources. This also suggest that, on the average, two of the four sources from each disease, more often than not, lead to positive

transfer.

Furthermore, while there was no significant difference in accuracy between TRL and MS-TRL with two sources, the reduction in performance, however, when the number of sources increased were apparent, especially when all sources were used for MS-TRL. The improvement over TRL accuracy by iTRL was also significant. Here too, the power of the significance dropped gradually as the number of sources (i.e., iterations) increased. Thus, more datasets and experiments might be need to estimate the inflection points of where number of sources change classification performance.

5.5.3 Discovery of Robust patterns with iTRL

Like tables 38 to 41 in section 5.4.4, tables 48 to 51 here represent robust rule patterns that were identified with iTRL. Recall that in iTRL, a rule pattern is said to be robust within or across a domain if it traversed (or survived) at least 50%—3 or more—of the datasets within a disease set. Thus only retained prior rules from source models S1, S2, and S3 were considered.

Majority of the robust patterns and variables as discovered with the iTRL framework from the breast (e.g., `ADIPOQ`, `COL11A1`, `ASPA`), colon (e.g., `HILPDA`, `CDH3`, `CNNM2`), lung (e.g., `AGER`, `EDNRB`, `AQP1`), and *mixed* (e.g., `ABLIM1`, `CALU`, `ABCG2`) set were identical with most of those captured with MS-TRL (see tables 38 to 41 and tables 48 to 51). See appendix F.2 for more details on robust rule patterns for the entire disease sets.

Like MS-TRL, the iTRL framework is sufficient for discovering robust rule patterns within and across homogeneous and heterogeneous domains respectively. Similarly, and as discussed in section 5.4.4, these patterns require further verification studies to affirm and/or falsify their utility for knowledge discovery within specified domains. For brevity, I will not expound on them any further as I have done in previous sections (see sections 5.3.8 and 5.4.4) with support from literature evidence. A future study, however, could compare and contrast the robust patterns that were discovered with the MS-TRL and iTRL frameworks.

Table 48: Examples of robust rule patterns that were discovered by iTRL on the combined breast cancer datasets. **S1, S2, S3:=**: Source datasets and also annotated in red font color, **T:=** Target dataset, which have also been marked in blue font color

S1:GEO7904, S2:GEO10780, S3:GEO15852, T:GEO29431
S2,T: 1. IF(BNIP3L=High) THEN (Class=CONTROL)
S3,T: 2. IF(ADH1C=High) THEN (Class=CONTROL)
S3,T: 3. IF(ACACB=High) THEN (Class=CONTROL)
S3,T: 4. IF(AOC3=High) THEN (Class=CONTROL)
S3,T: 5. IF(ANGPT1=High) THEN (Class=CONTROL)
S3,T: 6. IF(CCT3=High) THEN (Class=CASE)
S3,T: 7. IF(AOC3=Low) THEN (Class=CASE)
S3,T: 8. IF(ASPA=High) THEN (Class=CONTROL)
S3,T: 9. IF(ADH1B=High) THEN (Class=CONTROL)
S3,T: 10. IF(COL11A1=High) THEN (Class=CASE)
S3,T: 11. IF(ECT2=High) THEN (Class=CASE)
S3,T: 12. IF(CLDN5=High) THEN (Class=CONTROL)
S1:GEO42568, S2:GEO7904, S3:GEO10780, T:GEO15852
S3,T: 1. IF(ACIN1=High) THEN (Class=CONTROL)
S1,T: 2. IF(ADAR=High) THEN (Class=CASE)
S1,T: 3. IF(GPR157=High) THEN (Class=CASE)
S1:GEO7904, S2:GEO10780, S3:GEO29431, T:GEO42568
S2,T: 1. IF(CAP1=High)&(FHL1=Low) THEN (Class=CASE)
S3,T: 2. IF(GOS2=High) THEN (Class=CONTROL)
S2,T: 3. IF(ADIPOQ=Low) THEN (Class=CASE)
S2,T: 4. IF(ETNK1=High) THEN (Class=CASE)
S1,T: 5. IF(ADIPOQ=High)&(ADNP=Low) THEN (Class=CONTROL)

Table 49: Examples of robust rule patterns that were discovered by iTRL on the combined colon cancer datasets. **S1, S2, S3:=**: Source datasets and also annotated in red font color, **T:=** Target dataset, which have also been marked in blue font color

S1:GEO10715, S2:GEO23878, S3:GEO20916, T:GEO9348
S2,T: 1. IF(HILPDA=Low) THEN (Class=CONTROL)
S2,T: 2. IF(CDH3=High) THEN (Class=CASE)
S3,T: 3. IF(CDH3=Low) THEN (Class=CONTROL)
S2,T: 4. IF(CXCL12=High) THEN (Class=CONTROL)
S3,T: 5. IF(FABP6=High) THEN (Class=CASE)
S1:GEO9348, S2:GEO23878, S3:GEO24514 T:GEO10715
S3,T: 1. IF(DDX56=High) THEN (Class=CASE)
S3,T: 2. IF(DUS1L=High) THEN (Class=CASE)
S2,T: 3. IF(CXCL12=High) THEN (Class=CONTROL)
S3,T: 4. IF(CHP2=High) THEN (Class=CONTROL)
S3,T: 5. IF(CHGA=High) THEN (Class=CONTROL)
S1,T: 6. IF(CEBPB=Low) THEN (Class=CONTROL)
S2,T: 7. IF(BGN=High) THEN (Class=CASE)
S1,T: 8. IF(C4orf19=High) THEN (Class=CONTROL)
S3,T: 9. IF(CNNM2=High) THEN (Class=CONTROL)
S1:GEO9348, S2:GEO23878, S3:GEO20916 T:GEO24514
S3,T: 1. IF(CNNM2=High) THEN (Class=CONTROL)
S2,T: 2. IF(CBFB=Low) THEN (Class=CONTROL)
S3,T: 3. IF(ABCA8=Low) THEN (Class=CASE)
S2,T: 4. IF(CBFB=Low) THEN (Class=CONTROL)
S1,T: 5. IF(CA2=High) THEN (Class=CONTROL)
S3,T: 6. IF(CXCL2=Low) THEN (Class=CONTROL)
S2,T: 7. IF(ETV4=High) THEN (Class=CASE)
S3,T: 8. IF(CCT3=High) THEN (Class=CASE)
S2,T: 9. IF(CDH3=Low) THEN (Class=CONTROL)
S2,T: 10. IF(ADAMDEC1=High) THEN (Class=CONTROL)
S1,T: 11. IF(CEACAM7=High) THEN (Class=CONTROL)
S1,T: 12. IF(AKR1B10=High) THEN (Class=CONTROL)

Table 50: Examples of robust rule patterns that were discovered by iTRL on combined lung cancer datasets. **S1, S2, S3:=**: Source datasets and also annotated in red font color, **T:=** Target dataset, which have also been marked in blue font color

S1:GEO10072, S2:GEO19188, S3:GEO18842 T:GEO7670
S1,T: 1. IF(EDNRB=Low) THEN (Class=CASE)
S2,T: 2. IF(AQP1=Low) THEN (Class=CASE)
S3,T: 3. IF(GAPDH=High) THEN (Class=CASE)
S3,T: 4. IF(CLDN18=High) THEN (Class=CONTROL)
S1,T: 5. IF(ARHGAP6=Low) THEN (Class=CASE)
S3,T: 6. IF(CENPF=High) THEN (Class=CASE)
S2,T: 7. IF(ADRB2=Low) THEN (Class=CASE)
S2,T: 8. IF(AGER=High) THEN (Class=CONTROL)
S2,T: 9. IF(FABP4=Low) THEN (Class=CASE)
S1,T: 10. IF(CDH5=Low) THEN (Class=CASE)
S2,T: 11. IF(CITED2=Low) THEN (Class=CASE)
S2,T: 12. IF(CAV1=High) THEN (Class=CONTROL)
S2,T: 13. IF(ALOX5=Low) THEN (Class=CASE)
S2,T: 14. IF(ABCA8=Low) THEN (Class=CASE)
S2,T: 15. IF(CACYBP=High) THEN (Class=CASE)
S3,T: 16. IF(COX7A1=Low) THEN (Class=CASE)
S1,T: 17. IF(FRY=Low) THEN (Class=CASE)
S1,T: 18. IF(GPM6B=Low) THEN (Class=CASE)
S2,T: 19. IF(CELF2=Low) THEN (Class=CASE)

Table 51: Examples of robust rule patterns that were discovered by iTRL on combined set of randomly mixed cancer datasets. **S1, S2, S3:=**: Source datasets, **T:=** Target dataset, and color annotations (Brain := Red, Breast := Blue, Colon := Green, Lung := Purple, Prostate := Orange) denote the cancer type.

S1:GEO6956, S2:GEO7670, S3:GEO7904, T:GEO4412
S1,T: 1. IF(ATP5C1=Low) THEN (Class=CASE)
S2,T: 2. IF(ABLIM1=High) THEN (Class=CONTROL)
S2,T: 3. IF(CALU=High) THEN (Class=CASE)
S3,T: 4. IF(CEBPB=Low) THEN (Class=CONTROL)
S2,T: 5. IF(CCT6A=High) THEN (Class=CASE)
S1,T: 6. IF(ALG3=High)&(CLIC1=Low) THEN (Class=CASE)
S1:GEO4412, S2:GEO7904, S3:GEO7670, T:GEO6956
S3,T: 1. IF(CCT3=High) THEN (Class=CASE)
S3,T: 2. IF(CCL23=High) THEN (Class=CONTROL)
S3,T: 3. IF(FABP4=Low) THEN (Class=CASE)
S2,T: 4. IF(GNG11=High) THEN (Class=CONTROL)
S1:GEO6956, S2:GEO4412, S3:GEO7670, T:GEO7904
S2,T: 1. IF(ABCG2=Low) THEN (Class=CASE)
S2,T: 2. IF(ADORA2B=Low)&(ARHGEF6=Low) THEN (Class=CASE)
S2,T: 3. IF(ALPL=Low)&(ARHGEF6=Low) THEN (Class=CASE)
S3,T: 4. IF(ACTG2=Low)&(ANXA3=Low) THEN (Class=CASE)

5.5.4 Results summary - iTRL

Based on the experimental results discussed in this section, salient attributes about the iTRL framework that were revealed can be highlighted as follows:

1. The iTRL framework, for most cases, improves the accuracy of the baseline (i.e., more consistent). The magnitude of the improvement, i.e., positive/negative transfer, however, depends on the independent accuracies of sources as well as their ordering. Given a set of sources, where at least one can independently improve accuracy of the target, there exist a set of ordering involving these sources that will always lead to positive transfer. In addition, increasing the number of sources (or iterations) does not necessarily increase the magnitude or power of transfer.
2. Within a subspace of the data examples, which it covers, the iTRL, with the OnlyPriors variant, is more accurate than the Combo search. Given the entire data example, however, it performs poorly than the latter.
3. The iTRL Combo improves the baseline coverage by reducing the rate of abstentions. On the average, the rate of abstentions does not necessarily decrease with number of source or iterations. The OnlyPriors version, on the other hand, has relatively low coverage. The rate of abstentions increase with number of sources or iterations. Thus, only the Combo version improves baseline completeness.
4. Like MS-TRL, the iTRL framework is capable of capturing robust rules patterns within and across homogeneous and heterogeneous domains, respectively. Though further verification analysis of them, say in a wet lab, is required, literature evidence supports majority of them.

6.0 CONCLUSIONS

In this dissertation, I presented four novel frameworks, i.e., TRL-FM, KARL, MS-TRL, and iTRL, that combine background knowledge during search for predictive rule models from multiple sources of related gene expression data using transfer learning. I implemented them with four distinct algorithms, which provide significant extensions to an existing transfer rule learning method. They provide sufficient mechanisms to augment (e.g., tagging of sources, and knowledge extraction), combine and store (e.g., prior rules) domain knowledge for the transfer and use of robust rule patterns from multiple related datasets while learning a predictive model on the target. In addition, empirical results from an extensive evaluation on several gene expression data sets, reveal some key findings. First, they are more *complete*; combining information from multiple sources enables the rule model to describe more data examples. The more sources are incorporated into transfer rule model, the more expansive it becomes. Second, they are more *consistent*; more often than not, combining information from multiple sources improves prediction of a new data instance. The magnitude of the improvement, however, does not necessarily increase with amount of sources, but independent predictive performance of the sources as well as their ordering in the multi-source transfer framework. Third, by using the framework both domain-specific and domain-independent (general) robust rule patterns can be learned from appropriate homogeneous or heterogeneous biomedical datasets. Last, the methods developed herein provides generic frameworks that could be applied to other domains apart from biomedical data sets, to develop robust predictive rule models. These key findings thus confirm the overarching hypothesis of this work.

6.1 CONTRIBUTIONS

The advent of high-throughput genomics has led to the accumulation of biomedical data such as gene expression, are available through public repositories such as NCBI's GEO or EMBL-EBI's ArrayExpress. Meanwhile, the digitization of biomedical literature into repositories such as PubMed, have inspired the creation of curated knowledge bases like the Gene Ontology or Ingenuity® Knowledge Base. Pooling information from as many of these repositories as useful and integrating it with predicting modeling of similar biomedical data from multiple studies, could lead to models that are more robust.

This dissertation describes the development and evaluation of novel approaches for robust predictive modeling from multiple related biomedical data sources via transfer learning of classification rules. The new methods are able to avail pertinent information contained in several biomedical data sources to augment the knowledge discovery process in classification rules. The main contribution is the development of a knowledge extraction engine, which sits on an existing transfer rule learning framework, TRL, to facilitate information extraction, data preprocessing and transformations, and the abstraction of background knowledge into classification rules. Based on specific needs and use cases the multi-source transfer rule learning engine can be subdivided into four distinct, but related, frameworks presented herein:

TRL-FM implements a semi-automated algorithm, in a three-pronged approach, that abstract background knowledge into classification rules for knowledge transfer. First, it extracts pertinent domain knowledge from repositories such as the Gene Ontology. Second, it abstracts the information extracted into distinct ontology-based functional modules. Third, the functional modules are used as a bridge to map variables across multiple datasets to facilitate the generation of prior rules for transfer learning. While TRL requires identical variables across the source and target datasets for knowledge transfer to happen, TRL-FM goes a step further by availing relevant domain-knowledge that are contained in external sources to map non-identical variables for transfer. Though a group of variables may be described with non-identical symbols, they may belong to the same gene-family, biochemical pathway, or play several related roles in the same disease. The

ability to take cognizance of this underlying domain knowledge and incorporate them into predictive rule modeling enables TRL-FM to produce more informative and robust rules than TRL.

KARL implements a semi-automated algorithm, applies domain knowledge to define rule *interestingness*. In contemporary classification rule modeling, a rule is said to be interesting if it satisfies certain requirements, which are usually based on statistics estimated on the training data examples. Such measures, which are also *objective*, may not yield robust rule models, particularly in a noisy environment like biomedical data. To reduce the impact of such noise, background knowledge of the domain could be used to define and/or augment rule interestingness. Such measures are *subjective* since the definition of interestingness is at the discretion of a user. The key contribution of KARL is that it applies the subjective notion of interestingness to augment the knowledge discovery process. For proof of concept, it adopted cancer as a domain; and a rule is deemed interesting if it contained variables that are significantly associated to the hallmarks of cancer as contained in curated knowledge repository like IPA. First, it considers all variables in the target dataset that can also be found in the external source as containing evidence of association to the hallmarks. Second, the evidence is abstracted into a data structure called the functional lookup table. Third, information contained in the lookup table is used to abstract background knowledge into classification rules. Last, the background knowledge is used to augment knowledge discovery. The uniqueness about this method is that some of the prior rules may not have strong statistical support on the training data, but may contained a strong evidence of disease association from the literature. Meanwhile, majority of the infused background knowledge complements new rules to uncover hidden nuggets of robust rule patterns that are germane to the domain. Thus, by combining the background knowledge with new rules, the models generated by KARL are more expressive, robust, and have better predictive performance as compared to learning without background knowledge.

MS-TRL was built directly on top of TRL. Unlike TRL, which relies on one source model for prior rules, MS-TRL incorporates as much number of sources as useful to generate background knowledge. The key contribution here is the scale of possible prior rules. As

more transcriptomic data of related studies become available, it provides a mechanism that can involve as many sources as practicable. Depending on the characteristics of the source dataset, the performance of the model could be improved or degraded, following transfer learning. A single source transfer is thus a "hit or miss" endeavor. With multiple sources, it is possible to sift through the "good" and the "bad" before transfer. Another vital contribution of MS-TRL was that it applied a data transformation method (normalized linear transform) to align the source(s) and target datasets into a comparable numerical range before transfer. This mechanism enables discretization cut-off points or "whole rules" to be transferred from the sources. Given multiple related transcriptomic datasets, the algorithm first transforms the variables of the sources and target into a comparable numerical range. Second, it learns classification rule models on the preprocessed source datasets into source rule models. Third, it merges the source rule models into a unified set of background knowledge. Like TRL-FM and KARL, the background knowledge is then used to augment learning of the target rule model. Similarly, due to the increased number of sources it is able to generate rule models that are more expansive, robust, and improved predictive performance. MS-TRL provides a novel annotation of rules which is able to track rule patterns that co-occur in multiple models. When the datasets are homogeneous or common to the same type of disease, for instance, this mechanism could be used to discover robust patterns specific to the domain. On the other hand, if the datasets emanates from a heterogeneous domain, like a mix of different types of diseases, it could be used to also detect rule patterns that are general or independent of any particular domain. This feature is particularly useful for cross-domain studies such as "panomic". Last, MS-TRL implements a mechanism for updating rule-level statistics that could make it particularly useful for federated modeling. Instead of estimating the statistics of a transferred rule from scratch on a new dataset, the counts of its performance on related studies could be transferred, instead, to update its statistics on the new data, accordingly. This mechanism, which ameliorates *catastrophic forgetting*, improves efficiency of transfer rule learning, as well as the discovery of robust rule patterns. A rule that performs well on one dataset might perform badly on another. When such a rule is transferred, it may not meet the good rule criteria on the

target, and every prior information about it may be lost; in machine learning parlance, such a phenomenon is known as *catastrophic forgetting*.

iTRL applies a similar concept as MS-TRL, however, it does not merge all the source rule models into a unified set before transfer. Instead, it transfers one source model at a time, while updating rule statistics on the target, where applicable. This mechanism is particularly useful in scenarios where all the source datasets are not available at the same time. The target model can rather be updated upon arrival of new data. Meanwhile, it could also be used to discover robust rule patterns within or across domains. When a particular rule "survives" (or is retained) a number of N iterations, it could be deemed as robust. The main contribution of iTRL was to provide an avenue and mechanism for on-line learning, using the transfer rule learning framework.

Furthermore, KARL, MS-TRL, and iTRL implemented two mechanisms of rule space search: (1) and also the default, employs a quasi-parallel beam to combine background knowledge with new information to search for predictive rule models from multiple sources, and (2) combines only prior rules from multiple related models to search for predictive rule models. While models developed by the latter are less general and less robust than the former, they perform better within a subset of validation datasets. Thus, they could be used to specialize models on a subset of training data. In the event where data examples are not available for training new models, it provides an avenue for previously developed models to be combined for making inference.

6.2 LIMITATIONS

The results presented in this dissertation suggest that the multi-source transfer rule learning frameworks, i.e., TRL-FM, KARL, MS-TRL, and iTRL, are useful for combining background knowledge from multiple sources to discover robust predictive rule models. The conclusions should, however, be interpreted, bearing in mind, the following limitations:

(a) The frameworks were evaluated with microarray data sets that were downloaded from

NCBI GEO, where, predominantly, the assay platform are of the Affymetrix family. It should be desirable to validate them with data sets from other repositories (e.g., ArrayExpress, or TCGA) and/or platforms (e.g., Illumina). In addition, they could be validated on modern gene expression technologies like RNA sequencing (RNA-Seq).

(b) The discovery of robust rule patterns within homogeneous and heterogeneous datasets were premised on only five datasets. Although literature evidence supported majority of the discoveries, they cannot be wholly generalized per se, until validated on large number of datasets.

(c) Data transformation methods like discretization are known to cause information loss. In addition, **EBD** was the only discretization method used; other methods (e.g., Fayyad & Irani's MDL) could yield different results, as discretization, inherently, is a feature selector.

(d) The frameworks were founded on RL, which involves many parameters. Different parameter settings leads to different results. An example of this consequence was demonstrated in section 5.2.1, where different conflict-resolution methods produced different results.

(e) Several biomedical knowledge bases have been created that contain vital information on gene function, pathways, diseases, drugs, etc. The information contained in some of them (e.g., GO, KEGG, IPA) could overlap, yet they are diverse, in general. Thus relying solely on GO (i.e., TRL-FM) and IPA (i.e., KARL) to extract domain knowledge might limit the knowledge base for generating prior rules. Different repositories would result in different set of prior rules, which would subsequently result in different rule models.

6.3 FUTURE WORK

The experimental work as presented in this dissertation was intended as proof of concept to highlight the utility of applying transfer learning for combining background knowledge from multiple sources for predictive rule modeling of gene expression data. The observations and key findings may create avenues for extensions and directions for future work.

Simplification of rule models. Combining prior knowledge from multiple sources may lead to the generation of too many rules, especially when the number of source datasets are large. A relatively large set of rules may cause the model to be complex and less parsimonious [19]. Exploring an intelligent mechanism to reduce the rule size, while maintaining its predictive performance at a desirable minimum may be required. Previous work and preliminary results from TRL-FM have shown that some variables, though represented with different symbols, perform the same or similar functions within a domain. Thus, two or more rules that describe the same pattern (i.e., same *consequent*, similar variable values in *antecedent*, but different variables), but with different variables could be collapsed into a single rule representation. Such a single representation, for instance, could represent a pathway or biological process.

Incorporate background knowledge in rule confidence. Currently, all methods for estimating rule confidence are founded on rule-level statistic, which are calculated on the training data. We have learned from this exercise that the propensity of a rule pattern to occur could have strong support from a domain expert, knowledge base, and/or literature findings. Such a rule, however, may have less statistical support, and hence, low confidence on some data sets. Thus, there is a critical need for new approaches to compute rule confidence by combining evidence from background knowledge with statistics on the training data. A Bayesian approach, for instance, that could combine evidence from prior rules and literature to estimate rule confidence, could be explored.

Intelligent selection of prior models. Key findings from the experiments suggested that multi-source transfer rule learning methods statistically significantly outperforms learning without background knowledge. We also learned that increasing the number of sources does not necessarily correlate with positive transfer. This is because some sources improve learning, while others ruin it. Therefore, increasing the number of sources naively could be an exercise in futility. An intelligent approach is thus required to filter and select the "best" sources before transfer. One approach is to employ a validation set to pre-test all the source models before transfer. The ones whose accuracy exceed a certain threshold, say β, could be considered for transfer. In addition, the feasibility of genetic algorithms to address this challenge could be explored.

Intelligent ordering of prior models. We learned that iTRL is a viable tool for online learning using the multi-source transfer rule learning mechanism. Experimental results also revealed that the ordering of source models really matter for predictive performance. Given a set of source models, some particular orderings within the iterative process improve predictive performance, while some can ruin it. Devising an intelligent mechanism to get the ordering right is crucially needed. Combination of "good" source models at the beginning or tail-end of the iteration may significantly improve knowledge transfer. Identifying such good sources, as well as their right ordering, may not be trivial. The utility of stochastic methods like random walk could be explored for this open problem.

APPENDIX A

TRL++ MANUAL

```
NAME
    TRL++ - Program for transfer rule learning, variable selection
    and discretization, and data preprocessing.
SYNOPSIS
    java [JAVA_PARAMETERS] -jar TRLplus.jar
         -lp [LEARNING_PARAMETERS]
         -dp [DATA_PARAMETERS]
DESCRIPTION
    TRL++, is a toolkit for learning rule-based classification
    models from data. It implements several algorithms
    for learning classification rules--Classic RL, Transfer Rule
    Learning (TRL), Multiple Source Transfer Rule Learning (MS-TRL),
    Incremental Transfer Rule Learning (iTRL), and Knowledge Augmented
    Rule Learning (KARL). The algorithms have several learning
    parameters with sensible default values. The program can also
    transform input data in various ways before learning, or even
    without any learning.

    TRL's input is a set (or multiple) of training data instances, each
    specified in a data file (see DATA FILE FORMAT), where each
```

instance is a vector of values for the input variables, and a class value. The variables can be continuous or categorical. With the input it learns classification rule model, which comprises an unordered list of rules of the form:

IF <antecedent> THEN <consequent>

where the antecedent consists of a logical conjunction of one or more variable-value pairs (conditions), and the consequent is a prediction of the class variable. For example, a learned rule might be:

IF ((Age=High) AND (BloodPressure=Low)) THEN Class=Control

which means ‘‘if the variable *Age* is in the *High* range, and the variable *BloodPressure* is in the *Low* range, then predict that the data instance has the class value *Control.*’’ Values such as *Low* and *High* represent intervals of real numbers that result from discretizing the variables before learning. A rule is said to cover or match a data instance if each variable value of the instance is in the range specified in the rule antecedent. The classifier also includes an evidence gathering method for breaking ties when several rules match a query data instance but predict different classes.

The classic RL algorithm proceeds as a heuristic beam search through the space of rules from general to specific. Starting with all rules containing no variable-value pairs, it iteratively specializes the rules by adding conjuncts to the antecedent. It evaluates the rules, calculating a certainty factor value and other statistics for each rule. It re-inserts promising rules onto the beam, while removing

other rules. The beam is sorted by decreasing certainty factor value and is trimmed to a pre-defined length during each iteration. Beam search is used to limit the running time and space of the algorithm.

Multiple learned rules in an RL classifier may cover the same training instance. This is unlike most other classification rule and tree learning algorithms, which cover data without replacement, so that each data instance is covered by only one rule. With small sample size data sets, covering with replacement allows RL, and its extensions, to utilize more of the available evidence for each rule when computing the generalizability of the rule.

The extensions to RL (i.e., TRL, MS-TRL, iTRL, and KARL) also allows transfer learning, where some ‘‘prior’’ rules are learned on ‘‘source’’ data set(s) or domain-knowledge bases, and are then placed on the beam for learning on the ‘‘target’’ data set, while also learning a new set of rules on this data set.

DATA FILE FORMAT

Each data file comprises a table of rows (lines) and columns representing vectors of variable values. Example data file:

```
#ID  Age  Sex  Temperature  @Diagnosis
A42  22   F    37           Healthy
D25  35   M    39           Sick
...  ...  ...  ...          ...
```

In normally-oriented data files, each row represents a data instance vector and each column represents a variable of values for the instances; the first row is a header line specifying the names of the variables. However, the input file can be transposed, so that

rows represent variables and columns represent data instances; this must be specified using the ‘‘-tpf’’ option.

Columns can be separated by tabs or by commas (CSV). The separator must be the same throughout the file, and is assumed to be Tab unless "," occurs more frequently in the header line. The user can explicitly specify the separator by using the data parameters.

The data must contain exactly one class variable (output variable), and a number of input variables. The class variable is indicated by a ‘‘@’’ as the first character of the variable name. The class value of the first data instance that appears in the file is used in the predictive performance statistics in the output.

A data file may contain one ID input variable, indicated by a ‘‘#’’ as the first character of the variable name. TRL ignores the ID variable during learning but uses it to identify data instances in the output. If no ID variable is specified, the program uses data instance IDs ‘‘1’’, ‘‘2’’ and so on, namely the index of the data instance in the data file.

Each input variable can be continuous or categorical. A continuous variable is one whose values can be parsed as a numbers, such as ‘‘100’’ or ‘‘1.25’’ (without the quotes). Categorical variables have values such as ‘‘F’’ and ‘‘M’’. Before learning, TRL discretizes the continuous-valued variables using the specified discretizer. If you want the program to treat some numeric variable as categorical instead of continuous, and thus avoid discretizing it, you can add a single quote before each value. For example, instead of ‘‘100’’ (without the quotes), use ‘‘’100’’.

JAVA PARAMETERS

Because TRL is a java program, it must be run on a Java virtual machine. Java can take a number of parameters, which are described in the Java manual. In addition, it uses some of the Weka program libraries, so Weka must be installed on the system and the Java classpath must include the Weka classes (jar file). The classpath can be set in the CLASSPATH environment variable, or using the ‘‘-classpath’’ Java parameter. To provide enough memory on the Java virtual machine, use the ‘‘-Xmx’’ Java parameter. For example, ‘‘java -Xmx1000m’’.

LEARNING PARAMETERS

Learning parameters are command line attributes that bias the learning process of the algorithms, and are initiated by the flag ‘‘-lp’’. They can be specified in any order after the Java parameters and must precede the data parameters flag, ‘‘-dp’’, on the command line. The learning parameters, including their meanings, are presented below.

-cftype INDEX_VAL

Function to compute the certainty factor value for each rule.

0 Positive predictive value (default):

TP / (TP + FP)

1 Positive predictive value with Yates correction:

(TP + 0.05) / (TP + FP) if TP > FP,

(TP - 0.05) / (TP + FP) if TP < FP,

TP / (TP + FP) otherwise.

2 Positive predictive value, normalized for asymmetric class distributions:

1 if FP = 0

```
        0   if TP + FP = 0
        TP / (TP + FP * Pos / Neg)
3   Laplace estimate:
        (TP + 1) / (TP + FP + num_of_classes)
4   Laplace extended:
        (TP + k*m) / (TP + FP + k),
    where
        k = number of target values
        m = Pos / (Pos + Neg)
5   Laplace extended with bias for short rules:
        (TP + c * q) / (TP + FP + k),
    where
        c = 1 + number of conjuncts in the rule
        q = TP / (TP + TN)
6   F-Measure:
        2 * precision * recall / (precision +recall)
    where
        precision = TP / (TP + FP)
        recall    = TP / (TP + FN)
7   Laplace estimate, normalized for asymmetric class
    distributions:
        (TP + 1) / (TP + FP * Pos / Neg + num_of_classes)
8   P-Value, log likelihood ratio test:
        2 * (TP * log2(TPfreq/PosFreq) + \
             FP * log2(FPfreq/NegFreq))
    where
        TPfreq  = TP / (TP + FP)
        FPfreq  = FP / (TP + FP)
        PosFreq = Pos / (Pos + Neg)
        NegFreq = Neg / (Pos + Neg)
```

-inftype INDEX_VAL

The "inference type" or "evidence-gathering" function that is to be used during inference to make a prediction from a given data instance from the learned set of rules. It combines the predictions of all rules that match the given instance.

0 Weighted voting (default). Predict the highest-weighted class, where the weight of each class is the sum of certainty factors of rules predicting that class. If there is a tie, predicts class 0.

1 Maximum likelihood ratio

2 "Combine CF"

3 Lowest p-value: use the rule with the lowest p-value

4 Single best: use the rule with the highest certainty factor

5 Minimum weighted voting. Like weighted voting, but use only the highest k rules to calculate the weight of each class, where k is the minimum number of rules voting for any class.

6 Single best specific: use the rule with the highest worth (certainty divided by cost) and the highest number of conjuncts.

7 Most specific single best: use the rule with the most conjuncts among rules with the highest certainty factor.

8 Highest Coverage: use the rule with the highest coverage.

-mincf NUMBER

The minimum certainty factor value that any rule in the model will have. The default is 0.80.

-minconj NUMBER

The minimum number of conjuncts in any rule in the model. The default is 1.

-maxconj NUMBER

The maximum number of conjuncts in any rule in the model. The default is 5.

-specialize

If this option is specified, when a rule is added to the model, RL will also check if some specializations of this rule should also be added to the model. If the option is omitted (default), RL stops specialization of a rule once it is found to satisfy the search constraints.

-cover NUMBER

The minimum number of training examples that any rule in the model will cover. The default is 4.

-minTP DECIMAL

The minimum true positive rate that any rule in the model will have. The defalt is 0.05. Valid values are in the range [0, 1].

-maxFP DECIMAL

The maximum false positive rate that any rule in the model will have. This option is not set by default. Valid values are in the range [0, 1].

-indStr NUMBER

Inductive strengthening: the minimum number of previously uncovered examples that each new rule must cover. The default is 1. The

smaller this number, the larger the overlap of instances covered by different rules. Because RL covers data with replacement, using some non-zero inductive strengthening helps to learn a more generalizable model.

-beam WIDTH

The number of rules kept at any time to be specialized in the next iteration iteration. The default is 2500.

-cv NUM_FOLDS

Stratified cross-validation. If the option is not specified, no cross-validation is performed. In any case, a classifier is learned on all the training (target) data.

-d Discretize. The parameters are as described in PREPROCESSING PARAMETERS, but the discrete intervals for each variable are computed based on the training data only, then applied to the test data. If cross-validation is specified, the discretization is computed on the training subset separately for each fold.

TRANSFER LEARNING PARAMTERS

For the transfer learning algorithms to be invoked the ‘‘-tr’’ flag must first be specified, else the program runs RL by default. Below are the specific transfer learning parameters, including descriptions of their usage.

-tr INDEX_VAL

The type of transfer for prior rules. Prior rules are handled before any learning of new rules on the training data. The *INDEX_VAL* argument is a number, which signals the type of

transfer, and it can be denoted by either ‘‘1’’ or ‘‘2’’:

1 Whole rules (default)

Transfer each prior rule, including the exact variable-value pairs of both its antecedent and consequent. This ensures that the source and target (training) data have the same values for each variable.

2 Rule structure

Each prior rule is converted to a generalized structure where the variable values are removed, leaving only the variables in the LHS and the RHS. After transfer, this structure is converted into a set of rules, where for each variable all possible combinations of values in the target are considered.

-nocoverprior

Ignore the coverage of prior rules for inductive strengthening. That is, examples covered by prior rules will be considered not previously covered until they are covered by new rules.

-nopriorrulesspecialize

Do not specialize prior rules. If this option is omitted (default), prior rules are specialized on the beam.

-onlypriorrules

Perform heuristic rule space search with only prior rules. The default is a combo/dual beam search, where different beams are assigned to the prior and new rules for a side-by-side beam search.

-parallel

This flag invokes the main MS-TRL algorithm, where prior rules are generated from multiple source data sets in a parallel fashion.

Ensure that the ‘‘-src’’ data parameter (see DATA PARAMETERS) is flagged.

-serial

This flag invokes the iTRL algorithm, where transfer learning occurs in an incremental manner given multiple data sets. Ensure that the ‘‘-src’’ data parameter (see DATA PARAMETERS) is flagged.

-lookuptable FILENAME

This flag invokes the KARL algorithm, where prior rules are generated from abstracted evidence of domain-knowledge contained in the *FILENAME*. The *FILENAME* is a csv file, which contains domain variables and their evidence (from literature) of association with pertinent domain processes. Below is a snippet of a lookup table file which contains abstracted information from the domain, cancer.

GeneID	Findings	#Evidence	Function
COL1A1	-1	8	1
IGFBP2	+1	9	1
VEGFA	+1	31	2
CAV1	-1	24	3
...	...	...	...

In the example above, ‘‘GeneID’’ referes to the name of a domain variable (gene); ‘‘Findings’’ indicates the general consensus and evidence from literature on whether the variable decreases(-1), increases(+1), or affects (0) specific domain processes; ‘‘Function’’ denotes specific domain processes like cell death(1), cell invasion (2), or cell proliferation (3); and ‘‘#Evidence’’

indicates the number of literature references that attest to the functional evidence of the variable. Note that each lookup table file is specific to the training or target data set and must be processed outside the program.

PREPROCESSING PARAMETERS

These are optional parameters that specify operations to be performed on all the data before any rule learning. They can appear in any order after the "-lp" flag and before the "-dp" flag. Only operations specified will be performed.

-d DISC_MECHOD DISC_VALUE

Discretize using DISC_METHOD with specified parameter

PARAMETER	DISC_METHOD	PARAMETER2
0	GaussianU	Number of bins
1	EqualWidthU	Number of bins
2	EqualFreqU	Number of bins
3	OneR	Number of instances
4	ErrorBased	Max number of bins
5	D2S	(none - max number of bins is set to 8)
6	FayyadIraniMDL	Number of bins
7	HEBD	c structure prior (use value "1")
8	MODL	none?
9	EBD	lambda prior

Example: EBD (2011) discretization with default parameter:

-d 9 0.5

-r Remove trivial variables after discretization

```
    -chi NUM_OF_VARS_TO_SELECT
         Chi-squared variables selection: select the top
         NUM_VARS_TO_SELECT variables.

    -s SCALING_METHOD ...
         Scale each variable by the specified SCALING_METHOD in turn
             0  0-1 scaling
             1  Subtract local minimum
             2  Subtract global minimum
             3  Log2
             4  Square root
             5  Exponent 2
             6  Square
             7  Normalize to mean 0 and standard deviation 1

    -ctr
         Combine technical replicates. The samples must have the same name,
         with '#' next to it

DATA PARAMETERS
    Data parameters specify the input and output files and their format.
    The training data file is a mandatory data parameter and must be the
    last parameter. Data parameters must be preceded by the "-dp" flag.
    The ``-dp'' flag and the data parameters must appear after the
    ``-lp'' flag and any learning parameters, and after the.

    -itrncsv
         Training data file is comma-separated

    -itstcsv
```

Test data file is comma-separated

-c CSV_DATA_FILE

Convert the csv-delimited file to tab-delimited or vice versa.

-tpf DATA_FILE

Transpose the file data file; that is, make the rows be columns (variables) and columns be rows (instances).

-dtr TRAINING_DIRECTORY

Directory containing the training data files, one file for each training data instance. Each file contains two columns: variable and value. Within the directory files grouped by class folder; e.g. inside the training directory, there are two folders: "disease" and "control". There should be no trailing "/" in TRAINING_DIRECTORY name.

-tst TEST_FILE

Specify a test data file

-dtst TEST_DIRECTORY

Similar to -dtr.

-od OUTPUT_DIRECTORY

The output directory where to write the result files. The directory is automatically created if it does not already exist.

-o OUTPUT_DATA_FORMAT

csv Comma-separated values format. The default is tab-separated.

-rand SEED

Specifies a seed for creating random folds for running multiple runs of RL with cross-validation. SEED is an integer. On Unix-like systems and on Windows, a random integer is provided by the RANDOM environment variable. If this option is not specified, the default seed is 1.

-cmbf DIRECTORY

Combine the files in DIRECTORY. Each file represents one training example (such as a mass spectrum), and contains two comma-separated columns. The first column contains the names of the variables (such as M/Z values). The second column contains the values for those variables (e.g., intensity values).

-src

This parameter flags the source data file(s) for learning rules for transfer in TRL, MS-TRL, and iTRL frameworks. One (for TRL) or more (for MS-TRL or iTRL) files can be specified after this flag. Note that this flag is mandatory for the aforementioned transfer rule learning frameworks.

TRAINING_FILE

The training data file is specified as the last argument. This argument must ALWAYS be specified. For all transfer learning algorithms, this data file is designated as the target dataset.

OUTPUT

The program prints to standard output a log of its working that includes the the program parameters (and learning parameters), classifier learned, predictive performance statistics, starting time

```
    and total running time.

    For each rule, the log includes the following statistics: CF,
    CF/cost, p-value, true positive (TP) count, false positive (FP) count,
    and test TP and test FP.  These last two statistics are the number
    of test examples for which the rule was applied correctly (TP) or
    incorrectly (FP) when using the whole model. When applying the model,
    a rule may not fire even if it matches a test example, because of
    interaction with other rules. (See the discussion of evidence
    gathering under the  ‘‘-inftype’’ parameter.)

    The program also creates some output files containing any pre-processed
    data, rules learned on the whole training data set, prior rules learned
    from source(s), rules learned from each cross-validation fold
    (if cross-validation was used), predictions on the data instances used
    in validation, and the predictive performance of the model calculated
    from the predictions. The files are in the output directory, which by
    default is named as TRL_run_YYYY-MM-DD-hhmmss where the last part of
    the file name is the time when the program was run. A dedicated output
    directory can be specified using the ‘‘-od’’ parameter.

EXAMPLES
    Learn using classic RL with default parameters (including EBD
    discretization) and 10-fold cross-validation.
        java -jar TRL.jar -lp -cv 10 -d 9 0.5 -dp data.txt

    Train and test an RL model on a training and test data, respectively
        java -jar TRL.jar -lp -d 9 0.5 -dp -tst test-data.txt train-data.txt

    Transfer rule learning with whole-rule transfer after averaging
```

```
    technical replicates and 10-fold cross-validation:
        java -Xmx1300m -jar TRL.jar -LP -tr 1 -cv 10 -PPP -ctr\
        -DP -src source-data.txt target-data.txt

    Single source TRL with structure transfer
        java -jar TRL.jar -lp -tr 2 -d 9 0.5 \
        -dp -src source-data.txt target-data.txt

    Multiple source transfer rule learning (MS-TRL)
        java -jar TRL.jar -lp -tr 1 -parallel -d 9 0.5 \
        -dp -src source-data-1.txt source-data-2.txt target-data.txt

    Incremental transfer rule learning (iTRL)
        java -jar TRL.jar -lp -tr 1 -serial -d 9 0.5 \
        -dp -src source-data-1.txt source-data-2.txt target-data.txt

    Knowledge augmented rule learning (KARL)
        java -jar TRL.jar -lp -tr 2 -lookuptable LTable.csv -d 9 0.5 \
        -dp target-data.txt

    MS-TRL with only prior rules on 10 fold cross-validation
        java -jar TRL.jar -lp -tr 1 -parallel -onlypriorrules -cv 10 -d \
        9 0.5 -dp -src source-data-1.txt source-data-2.txt target-data.txt

AUTHORS
    Jonathan Lustgarten, 2006 - 2009
    Philip Ganchev, 2009 - 2010
    Jeya Balaji Balasubramanian, 2011 - 2015
    Henry Ato Ogoe, 2014 - 2016
```

APPENDIX B

TRL - SUPPLEMENTARY

Table 52: Classification performance of TRL Combo on breast cancer set

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**	**Ab (%)**
GEO42568	GEO10780	47.619	100.000	73.810	88.108	0.000
GEO7904	GEO10780	46.341	100.000	73.171	87.568	0.541
GEO29431	GEO10780	45.000	100.000	72.500	87.027	1.081
GEO15852	GEO10780	40.000	100.000	70.000	85.946	1.081
GEO10780	GEO15852	86.047	83.721	84.884	84.884	0.000
GEO42568	GEO15852	88.372	83.721	86.047	86.047	0.000
GEO7904	GEO15852	86.047	83.721	84.884	84.884	0.000
GEO29431	GEO15852	86.047	86.047	86.047	86.047	0.000
GEO15852	GEO29431	100.000	83.333	91.667	96.970	0.000
GEO10780	GEO29431	90.741	100.000	95.370	90.909	1.515
GEO7904	GEO29431	100.000	75.000	87.500	95.455	0.000
GEO42568	GEO29431	100.000	83.333	91.667	96.970	0.000
GEO7904	GEO42568	100.000	76.471	88.235	96.694	0.000
GEO15852	GEO42568	100.000	76.471	88.235	96.694	0.000
GEO29431	GEO42568	100.000	82.353	91.176	96.694	0.826
GEO10780	GEO42568	96.154	88.235	92.195	95.041	0.000
GEO10780	GEO7904	90.698	47.368	69.033	77.419	0.000
GEO29431	GEO7904	90.698	42.105	66.401	75.806	0.000
GEO15852	GEO7904	90.698	52.632	71.665	79.032	0.000
GEO42568	GEO7904	95.238	47.368	71.303	79.032	1.613

Table 53: Classification performance of TRL Combo on colon cancer set

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**	**Ab (%)**
GEO23878	GEO10715	78.947	60.000	69.474	70.000	3.333
GEO24514	GEO10715	73.684	70.000	71.842	70.000	3.333
GEO20916	GEO10715	83.333	55.556	69.444	66.667	10.000
GEO9348	GEO10715	83.333	50.000	66.667	63.333	13.333
GEO10715	GEO20916	100.000	97.059	98.529	97.143	1.429
GEO9348	GEO20916	100.000	94.118	97.059	97.143	0.000
GEO24514	GEO20916	97.143	100.000	98.571	97.143	1.429
GEO23878	GEO20916	100.000	100.000	100.000	100.000	0.000
GEO20916	GEO23878	94.286	100.000	97.143	96.610	0.000
GEO24514	GEO23878	100.000	100.000	100.000	100.000	0.000
GEO9348	GEO23878	97.143	100.000	98.571	93.220	5.085
GEO10715	GEO23878	100.000	86.957	93.478	93.220	1.695
GEO23878	GEO24514	100.000	86.667	93.333	95.918	0.000
GEO10715	GEO24514	100.000	86.667	93.333	95.918	0.000
GEO9348	GEO24514	100.000	86.667	93.333	95.918	0.000
GEO20916	GEO24514	100.000	93.333	96.667	97.959	0.000
GEO20916	GEO9348	100.000	100.000	100.000	100.000	0.000
GEO24514	GEO9348	100.000	91.667	95.833	98.780	0.000
GEO23878	GEO9348	100.000	91.667	95.833	98.780	0.000
GEO10715	GEO9348	100.000	66.667	83.333	92.683	2.439

Table 54: Classification performance of TRL Combo on lung cancer set

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**	**Ab (%)**
GEO19188	GEO10072	100.000	91.837	95.918	96.262	0.000
GEO18842	GEO10072	100.000	93.750	96.875	96.262	0.935
GEO7670	GEO10072	100.000	93.878	96.939	96.262	0.935
GEO19804	GEO10072	100.000	95.918	97.959	98.131	0.000
GEO10072	GEO18842	100.000	100.000	100.000	98.901	1.099
GEO19188	GEO18842	100.000	93.333	96.667	96.703	0.000
GEO19804	GEO18842	100.000	97.778	98.889	98.901	0.000
GEO7670	GEO18842	97.826	93.333	95.580	95.604	0.000
GEO10072	GEO19188	96.703	93.846	95.275	95.513	0.000
GEO18842	GEO19188	96.703	93.846	95.275	95.513	0.000
GEO19804	GEO19188	96.703	93.846	95.275	95.513	0.000
GEO7670	GEO19188	96.703	93.846	95.275	95.513	0.000
GEO7670	GEO19804	96.667	95.000	95.833	95.833	0.000
GEO18842	GEO19804	96.610	93.333	94.972	94.167	0.833
GEO10072	GEO19804	95.000	95.000	95.000	95.000	0.000
GEO19188	GEO19804	96.667	91.667	94.167	94.167	0.000
GEO19804	GEO7670	94.872	96.296	95.584	95.455	0.000
GEO10072	GEO7670	94.872	96.296	95.584	95.455	0.000
GEO18842	GEO7670	92.308	92.593	92.450	92.424	0.000
GEO19188	GEO7670	94.872	88.889	91.880	92.424	0.000

Table 55: Classification performance of TRL Combo on prostate cancer set

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**	**Ab (%)**
GEO32448	GEO17951	83.824	89.706	86.765	86.131	0.730
GEO46602	GEO17951	88.060	90.909	89.484	86.861	2.920
GEO6956	GEO17951	85.075	84.058	84.566	83.942	0.730
GEO82188	GEO17951	88.235	86.765	87.500	86.861	0.730
GEO6956	GEO32448	97.368	87.500	92.434	90.000	2.500
GEO17951	GEO32448	97.500	90.000	93.750	93.750	0.000
GEO82188	GEO32448	90.000	87.500	88.750	88.750	0.000
GEO46602	GEO32448	94.737	87.500	91.118	88.750	2.500
GEO32448	GEO46602	100.000	92.857	96.429	98.000	0.000
GEO6956	GEO46602	97.222	85.714	91.468	94.000	0.000
GEO17951	GEO46602	100.000	71.429	85.714	92.000	0.000
GEO82188	GEO46602	97.222	78.571	87.897	92.000	0.000
GEO17951	GEO6956	95.652	50.000	72.826	85.393	0.000
GEO82188	GEO6956	92.754	45.000	68.877	82.022	0.000
GEO32448	GEO6956	98.551	55.000	76.775	88.764	0.000
GEO46602	GEO6956	98.551	55.000	76.775	88.764	0.000
GEO17951	GEO82188	89.231	91.549	90.390	90.441	0.000
GEO46602	GEO82188	89.231	91.549	90.390	90.441	0.000
GEO6956	GEO82188	90.769	90.141	90.455	90.441	0.000
GEO32448	GEO82188	89.062	95.775	92.419	91.912	0.735

Table 56: Classification performance of TRL OnlyPriors on breast cancer set

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**	**Ab (%)**
GEO42568	GEO10780	80.769	80.000	80.385	35.135	56.216
GEO7904	GEO10780	83.333	50.000	66.667	23.243	62.162
GEO29431	GEO10780	100.000	0.000	50.000	1.081	98.378
GEO15852	GEO10780	88.571	82.301	85.436	67.027	20.000
GEO10780	GEO15852	45.161	89.474	67.317	55.814	19.767
GEO42568	GEO15852	94.737	86.486	90.612	79.070	12.791
GEO7904	GEO15852	83.333	94.737	89.035	76.744	13.953
GEO29431	GEO15852	66.667	94.286	80.476	43.023	52.326
GEO15852	GEO29431	100.000	58.333	79.167	90.909	1.515
GEO10780	GEO29431	70.270	90.000	80.135	53.030	28.788
GEO7904	GEO29431	100.000	63.636	81.818	92.424	1.515
GEO42568	GEO29431	100.000	81.818	90.909	93.939	3.030
GEO7904	GEO42568	100.000	68.750	84.375	95.041	0.826
GEO15852	GEO42568	99.038	70.588	84.813	95.041	0.000
GEO29431	GEO42568	98.901	82.353	90.627	85.950	10.744
GEO10780	GEO42568	74.359	100.000	87.179	61.983	21.488
GEO10780	GEO7904	15.385	33.333	24.359	6.452	69.355
GEO29431	GEO7904	90.476	33.333	61.905	37.097	46.774
GEO15852	GEO7904	84.375	41.667	63.021	51.613	29.032
GEO42568	GEO7904	86.842	40.000	63.421	62.903	14.516

Table 57: Classification performance of TRL OnlyPriors on colon cancer set

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**	**Ab (%)**
GEO23878	GEO10715	80.000	71.429	75.714	56.667	26.667
GEO24514	GEO10715	72.727	77.778	75.253	50.000	33.333
GEO20916	GEO10715	100.000	66.667	83.333	26.667	70.000
GEO9348	GEO10715	50.000	50.000	50.000	6.667	86.667
GEO10715	GEO20916	88.462	87.500	87.981	62.857	28.571
GEO9348	GEO20916	96.552	90.323	93.437	80.000	14.286
GEO24514	GEO20916	93.548	90.476	92.012	68.571	25.714
GEO23878	GEO20916	91.667	93.548	92.608	88.571	4.286
GEO20916	GEO23878	94.286	95.833	95.060	94.915	0.000
GEO24514	GEO23878	93.939	86.957	90.448	86.441	5.085
GEO9348	GEO23878	71.429	93.333	82.381	40.678	50.847
GEO10715	GEO23878	100.000	70.588	85.294	71.186	20.339
GEO23878	GEO24514	96.970	73.333	85.152	87.755	2.041
GEO10715	GEO24514	76.471	88.889	82.680	42.857	46.939
GEO9348	GEO24514	0.000	100.000	50.000	22.449	69.388
GEO20916	GEO24514	100.000	92.857	96.429	95.918	2.041
GEO20916	GEO9348	94.286	91.667	92.976	93.902	0.000
GEO24514	GEO9348	100.000	75.000	87.500	96.341	0.000
GEO23878	GEO9348	100.000	100.000	100.000	100.000	0.000
GEO10715	GEO9348	91.489	62.500	76.995	58.537	32.927

Table 58: Classification performance of TRL OnlyPriors on lung cancer set

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**	**Ab (%)**
GEO19188	GEO10072	98.276	83.673	90.975	91.589	0.000
GEO18842	GEO10072	92.000	88.235	90.118	71.028	21.495
GEO7670	GEO10072	98.214	93.750	95.982	93.458	2.804
GEO19804	GEO10072	96.296	91.304	93.800	87.850	6.542
GEO10072	GEO18842	100.000	100.000	100.000	97.802	2.198
GEO19188	GEO18842	100.000	87.805	93.902	90.110	4.396
GEO19804	GEO18842	97.826	100.000	98.913	95.604	3.297
GEO7670	GEO18842	95.238	93.023	94.131	87.912	6.593
GEO10072	GEO19188	95.604	93.750	94.677	94.231	0.641
GEO18842	GEO19188	88.506	86.538	87.522	78.205	10.897
GEO19804	GEO19188	97.802	93.846	95.824	96.154	0.000
GEO7670	GEO19188	96.703	90.769	93.736	94.231	0.000
GEO7670	GEO19804	93.220	91.228	92.224	89.167	3.333
GEO18842	GEO19804	90.196	85.366	87.781	67.500	23.333
GEO10072	GEO19804	93.333	91.228	92.281	90.000	2.500
GEO19188	GEO19804	96.667	71.186	83.927	83.333	0.833
GEO19804	GEO7670	92.308	96.296	94.302	93.939	0.000
GEO10072	GEO7670	94.737	96.296	95.517	93.939	1.515
GEO18842	GEO7670	84.211	91.667	87.939	81.818	6.061
GEO19188	GEO7670	100.000	88.462	94.231	93.939	1.515

Table 59: Classification performance of TRL OnlyPriors on prostate cancer set

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**	**Ab (%)**
GEO32448	GEO17951	74.242	88.710	81.476	75.912	6.569
GEO46602	GEO17951	97.778	74.194	85.986	48.905	44.526
GEO6956	GEO17951	90.625	55.000	72.812	66.423	9.489
GEO82188	GEO17951	88.235	71.212	79.724	78.102	2.190
GEO6956	GEO32448	93.939	50.000	71.970	51.250	33.750
GEO17951	GEO32448	94.737	76.316	85.526	81.250	5.000
GEO82188	GEO32448	81.579	83.784	82.681	77.500	6.250
GEO46602	GEO32448	86.957	72.414	79.685	51.250	35.000
GEO32448	GEO46602	100.000	92.857	96.429	98.000	0.000
GEO6956	GEO46602	100.000	83.333	91.667	90.000	6.000
GEO17951	GEO46602	100.000	50.000	75.000	86.000	0.000
GEO82188	GEO46602	94.444	85.714	90.079	92.000	0.000
GEO17951	GEO6956	83.871	47.368	65.620	68.539	8.989
GEO82188	GEO6956	76.923	47.368	62.146	66.292	5.618
GEO32448	GEO6956	83.636	70.588	77.112	65.169	19.101
GEO46602	GEO6956	84.615	69.231	76.923	34.831	56.180
GEO17951	GEO82188	83.871	85.507	84.689	81.618	3.676
GEO46602	GEO82188	92.308	52.941	72.624	39.706	46.324
GEO6956	GEO82188	88.710	72.131	80.420	72.794	9.559
GEO32448	GEO82188	86.885	91.176	89.031	84.559	5.147

```
1: function TRL(p_rules, D, C)
2:     ▷ p_rules : a set of prior rules
3:     ▷ D : a set of training examples
4:     ▷ C : user specified constraints for rule learning
5:     interesting_patterns ← ∅
6:     new_beam ← p_rules ∪ {∅ ⇒ class_1, ∅ ⇒ class_2, ...}
7:     beam ← ∅
8:     while new_beam ≠ ∅ do
9:         beam ← new_beam
10:        new_beam ← ∅
11:        for all rule ∈ beam do
12:            S ← SPECIALIZE(rule)
13:            for all s ∈ S do
14:                if ISRULEINTERESTING(s, IC, D) then
15:                    ▷ IC : user specified interestingness criteria
16:                    interesting_patterns ← interesting_patterns ∪ s
17:                end if
18:                if ISGOODRULE(s, C, D) then
19:                    new_beam ← new_beam ∪ s
20:                end if
21:            end for
22:        end for
23:    end while
24:    return beam
25: end function
```

Figure 30: Pseudocode for a heuristic rule-space search given prior rules, TRL

APPENDIX C

TRL-FM - SUPPLEMENTARY

Table 60: Description of datasets for TRL-FM experiments

Disease	Author	Year	Platform	Sample Size (T/N)	Source
Prostate Cancer	Singh	2002	HG-U95Av2	102 (52/50)	broad.mit.edu
	Lapointe	2004	cDNA	103 (62/41)	GSE3933
	Wallace	2008	HGU133A2	89 (69/20)	GSE6956
	Nanni	2006	HG-U133A	30 (23/7)	GSE3868
	Varambally	2005	HG-U133 Plus 2	13 (7/6)	GSE3325
	Welsh	2001	HG-U95A	34 (25/9)	public.gnf.org
	Yu	2004	HG-U95Av2	83 (65/18)	GSE6919
Brain Cancer	Freije	2004	HG-U133A,B	85 (59/26)	GSE4412
	Phillips	2006	HG-U133A,B	100 (76/24)	GSE4271
	Sun	2006	HG-UI33 Plus	100 (81/19)	GSE4290
	Petalidis	2008	HG-U133A	58 (39/19)	GSE1993
	Gravendeel	2009	HG-U133 Plus 2	175 (159/16)	GSE16011
	Paugh	2010	HG-U133 Plus 2	42 (33/9)	GSE19578
	Yamanaka	2006	Agilent	29 (22/7)	GSE4381
Lung Disease Studies (IPF)	Pardo	2005	Codelink	24 (13/11)	GSE2052
	Yang	2007	Agilent 43K	29 (20/9)	GSE5774
	Konishi	2009	Agilent 4x44K	38 (23/15)	GSE10667
	KangA	2011	Agilent 4x44K	63 (52/11)	Dr. Kaminski
	KangB	2011	Agilent 8x60K	96 (75/21)	Dr. Kaminski
	Larsson	2008	HG-U133 Plus 2	12 (6/6)	GSE11196
	Emblom	2010	cDNA	58 (38/20)	GSE17978

Table 61: Meta-analysis approach to integrate datasets into single disease-specific matrix. First, the common genes that occur among all datasets of a specific disease (e.g., IPF) are determined. Second, using the AW Fisher method [86], all candidate genes that are differentially expressed (DEGs) in one or more studies are determined. Last, disease-specific datasets are merged into a single matrix according DEGs. *T = Tumor; N = Normal

Cancer Type	**Dataset**	**Samples (T/N)***	**Genes**	**Common Genes**	**DEGs**	**Merged Matrix**
	Singh	102 (52/50)	9700			
	Lapointe	103 (62/41)	13579			
	Wallace	89 (69/20)	14704			
Prostate	Nanni	30 (23/7)	14713	6940	2869	454 X 2869
	Varambally	13 (7/6)	33727			
	Welsh	34 (25/9)	9700			
	Yu	83 (65/18)	9700			
	Freije	85 (59/26)	28168			
	Phillips	100 (76/24)	28168			
	Sun	100 (81/19)	33675			
Brain	Petalidis	58 (39/19)	14713	6019	1707	589 X 1707
	Gravendeel	175 (159/16)	17332			
	Paugh	42 (33/9)	19738			
	Yamanaka	29 (22/7)	12043			
	Pardo	24 (13/11)	8653			
	Yang	29 (20/9)	17198			
	Konishi	38 (23/15)	19749			
IPF	KangA	63 (52/11)	19614	5481	2455	320 X 2455
	KangB	96 (75/21)	22627			
	Larsson	12 (6/6)	16123			
	Emblom	58 (38/20)	16679			

Table 62: A batch-effect removal method (BERM) was used to merge disease-specific datasets into a single matrix. First, the common genes that occur among all datasets of a specific disease (e.g., IPF) are determined. Subsequently, disease-specific datasets are merged into a single data matrix with a BERM. The BERM used was COMBAT. Note that several BERM have been proposed, and the choice of COMBAT was arbitrary. *T = Tumor; N = Normal

Cancer Type	**Dataset**	**Samples (T/N)***	**Genes**	**Common Genes**	**BERM Merged**
	Singh	102 (52/50)	9700		
	Lapointe	103 (62/41)	13579		
	Wallace	89 (69/20)	14704		
Prostate	Nanni	30 (23/7)	14713	6940	454 X 6940
	Varambally	13 (7/6)	33727		
	Welsh	34 (25/9)	9700		
	Yu	83 (65/18)	9700		
	Freije	85 (59/26)	28168		
	Phillips	100 (76/24)	28168		
	Sun	100 (81/19)	33675		
Brain	Petalidis	58 (39/19)	14713	6019	589 X 6019
	Gravendeel	175 (159/16)	17332		
	Paugh	42 (33/9)	19738		
	Yamanaka	29 (22/7)	12043		
	Pardo	24 (13/11)	8653		
	Yang	29 (20/9)	17198		
	Konishi	38 (23/15)	19749		
IPF	KangA	63 (52/11)	19614	5481	320 X 5481
	KangB	96 (75/21)	22627		
	Larsson	12 (6/6)	16123		
	Emblom	58 (38/20)	16679		

APPENDIX D

KARL - SUPPLEMENTARY

D.1 CLASSIFICATION PERFORMANCE

Table 63: Mean classification (cross-validation) performance for KARL using Combo search. **SN** = Sensitivity, **SP** = Specificity, **BACC** = Balanced Accuracy, **Acc** = Accuracy, **AccAb** = Accuracy including abstentions, **Ab** = Abstentions

Dataset	**SN**	**SP**	**BACC**	**Acc**	**AccAb**	**Ab**	**Ab (%)**
GEO16011	95.60	18.75	57.17	88.57	88.57	0	0.00
GEO1993	92.31	73.68	83.00	86.21	86.21	0	0.00
GEO4271	100.00	8.33	54.17	78.00	78.00	0	0.00
GEO4290	96.30	31.58	63.94	84.00	84.00	0	0.00
GEO4412	93.22	53.85	73.53	81.18	81.18	0	0.00
GEO10780	42.50	100.00	71.25	87.43	86.49	2	1.08
GEO15852	83.72	86.05	84.88	84.88	84.88	0	0.00
GEO29431	100.00	100.00	100.00	100.00	100.00	0	0.00
GEO42568	99.04	76.47	87.76	95.87	95.87	0	0.00
GEO7904	90.70	52.63	71.67	79.03	79.03	0	0.00
GEO10715	83.33	72.73	78.03	79.31	76.67	1	3.33
GEO20916	100.00	100.00	100.00	100.00	100.00	0	0.00
GEO23878	100.00	100.00	100.00	100.00	100.00	0	0.00
GEO24514	100.00	86.67	93.33	95.92	95.92	0	0.00
GEO9348	100.00	100.00	100.00	100.00	98.78	1	1.22
GEO10072	100.00	91.84	95.92	96.26	96.26	0	0.00
GEO18842	100.00	93.33	96.67	96.70	96.70	0	0.00
GEO19188	96.70	90.77	93.74	94.23	94.23	0	0.00
GEO19804	96.67	95.00	95.83	95.83	95.83	0	0.00
GEO7670	94.87	85.19	90.03	90.91	90.91	0	0.00
GEO17951	89.71	94.20	91.95	91.97	91.97	0	0.00
GEO32448	92.50	85.00	88.75	88.75	88.75	0	0.00
GEO46602	100.00	92.86	96.43	98.00	98.00	0	0.00
GEO6956	95.65	70.00	82.83	89.89	89.89	0	0.00
GEO82188	92.31	91.43	91.87	91.85	91.18	1	0.74

Table 64: Mean classification (cross-validation) performance for KARL using only priors rules for search. **SN** = Sensitivity, **SP** = Specificity, **BACC** = Balanced Accuracy, **Acc** = Accuracy, **AccAb** = Accuracy including abstentions, **Ab** = Abstentions

Dataset	**SN**	**SP**	**BACC**	**Acc**	**AccAb**	**Abs**	**Abs (%)**
GEO16011	76.87	68.75	72.81	76.07	70.86	12	6.86
GEO1993	86.84	73.68	80.26	82.46	81.03	1	1.72
GEO4271	92.00	37.50	64.75	78.79	78.00	1	1.00
GEO4290	92.41	47.37	69.89	83.67	82.00	2	2.00
GEO4412	91.23	69.23	80.23	84.34	82.35	2	2.35
GEO10780	100.00	0.00	50.00	25.00	1.08	177	95.68
GEO15852	88.37	87.81	88.09	88.10	86.05	2	2.33
GEO29431	100.00	72.73	86.36	95.16	89.39	4	6.06
GEO42568	100.00	50.00	75.00	95.58	89.26	8	6.61
GEO7904	87.50	60.00	73.75	80.00	70.97	7	11.29
GEO10715	62.50	90.00	76.25	77.78	46.67	12	40.00
GEO20916	100.00	100.00	100.00	100.00	95.71	3	4.29
GEO23878	100.00	100.00	100.00	100.00	86.44	8	13.56
GEO24514	100.00	86.67	93.33	95.92	95.92	0	0.00
GEO9348	100.00	91.67	95.83	98.77	97.56	1	1.22
GEO10072	100.00	95.92	97.96	98.11	97.20	1	0.94
GEO18842	100.00	97.56	98.78	98.84	93.41	5	5.50
GEO19188	97.75	86.00	91.88	93.53	83.33	17	10.90
GEO19804	91.67	93.33	92.50	92.50	92.50	0	0.00
GEO7670	97.37	94.74	96.05	96.49	83.33	9	13.64
GEO17951	89.71	92.65	91.18	91.18	90.51	1	0.73
GEO32448	82.05	87.18	84.62	84.62	82.50	2	2.50
GEO46602	100.00	91.67	95.83	97.92	94.00	2	4.00
GEO6956	66.67	80.00	73.33	70.59	53.93	21	23.60
GEO82188	87.69	90.14	88.92	88.97	88.97	0	0.00

Table 65: Mean positive coverage for KARL with combo search on models on 10-Fold cross-validation. **ConMin** = minimum positive coverage for controls, **ConMax** = maximum positive coverage for controls, **ConMdn** = median positive coverage for controls, **CaseMin** = minimum positive coverage for cases, **CaseMax** = maximum positive coverage for cases, **CaseMdn** = median positive coverage for cases

Dataset	**ConMin**	**ConMax**	**ConMdn**	**CaseMin**	**CaseMax**	**CaseMdn**
GEO16011	0.395	0.695	0.548	0.549	0.790	0.643
GEO1993	0.538	0.778	0.681	0.507	0.761	0.657
GEO4271	0.329	0.654	0.495	0.430	0.820	0.618
GEO4290	0.327	0.702	0.529	0.453	0.745	0.584
GEO4412	0.342	0.817	0.526	0.375	0.825	0.580
GEO10780	0.497	0.836	0.649	0.386	0.601	0.471
GEO15852	0.421	0.858	0.739	0.362	0.795	0.526
GEO29431	1.000	1.000	1.000	1.000	1.000	1.000
GEO42568	0.889	0.902	0.889	0.948	0.981	0.971
GEO7904	0.433	0.603	0.521	0.493	0.747	0.625
GEO10715	0.667	0.909	0.813	0.848	0.959	0.892
GEO20916	1.000	1.000	1.000	1.000	1.000	1.000
GEO23878	1.000	1.000	1.000	1.000	1.000	1.000
GEO24514	0.926	0.948	0.937	0.731	0.997	0.935
GEO9348	1.000	1.000	1.000	1.000	1.000	1.000
GEO10072	1.000	1.000	1.000	1.000	1.000	1.000
GEO18842	1.000	1.000	1.000	1.000	1.000	1.000
GEO19188	0.535	0.959	0.662	0.716	0.956	0.898
GEO19804	0.481	0.957	0.730	0.600	0.939	0.724
GEO7670	0.799	0.983	0.881	0.832	0.974	0.930
GEO17951	0.324	0.891	0.481	0.472	0.884	0.665
GEO32448	0.381	0.817	0.611	0.350	0.792	0.533
GEO46602	1.000	1.000	1.000	1.000	1.000	1.000
GEO6956	0.411	0.822	0.608	0.565	0.820	0.686
GEO82188	0.354	0.850	0.585	0.363	0.875	0.569

Table 66: Mean positive coverage for KARL using search with only prior rules on models on 10-Fold cross-validation. **ConMin** = minimum positive coverage for controls, **ConMax** = maximum positive coverage for controls, **ConMdn** = median positive coverage for controls, **CaseMin** = minimum positive coverage for cases, **CaseMax** = maximum positive coverage for cases, **CaseMdn** = median positive coverage for cases

Dataset	**ConMin**	**ConMax**	**ConMdn**	**CaseMin**	**CaseMax**	**CaseMdn**
GEO16011	0.418	0.723	0.581	0.361	0.692	0.510
GEO1993	0.649	0.790	0.734	0.629	0.889	0.766
GEO4271	0.560	0.695	0.615	0.545	0.784	0.670
GEO4290	0.479	0.702	0.611	0.151	0.749	0.539
GEO4412	0.501	0.726	0.624	0.476	0.800	0.673
GEO10780	0.000	0.000	0.000	0.109	0.109	0.109
GEO15852	0.718	0.827	0.762	0.646	0.873	0.738
GEO29431	0.990	0.990	0.990	1.000	1.000	1.000
GEO42568	0.252	0.292	0.272	0.932	1.000	0.983
GEO7904	0.439	0.632	0.491	0.481	0.661	0.606
GEO10715	0.677	0.828	0.747	0.500	0.512	0.506
GEO20916	1.000	1.000	1.000	1.000	1.000	1.000
GEO23878	0.828	0.904	0.866	1.000	1.000	1.000
GEO24514	0.956	0.956	0.956	0.915	0.967	0.941
GEO9348	1.000	1.000	1.000	1.000	1.000	1.000
GEO10072	1.000	1.000	1.000	1.000	1.000	1.000
GEO18842	1.000	1.000	1.000	1.000	1.000	1.000
GEO19188	0.485	0.657	0.595	0.734	0.940	0.812
GEO19804	0.830	0.924	0.895	0.717	0.900	0.848
GEO7670	0.575	0.695	0.633	0.795	0.952	0.880
GEO17951	0.572	0.792	0.669	0.634	0.874	0.825
GEO32448	0.589	0.856	0.721	0.514	0.758	0.592
GEO46602	0.976	0.984	0.980	1.000	1.000	1.000
GEO6956	0.450	0.811	0.639	0.143	0.497	0.336
GEO82188	0.584	0.814	0.711	0.547	0.810	0.702

D.2 ROBUST RULE PATTERNS VIA KARL

D.2.1 BRAIN CANCER

```
=== GEO1993 ===
Pr 2. ((COL4A2 = -inf..8.226)) ==> (@Class = CONTROL)
   CF=0.933, PV=6.71939E-7, TP=13, FP=0, Pos=19, Neg=39
Pr 8. ((COL6A3 = 6.661..inf)) ==> (@Class = CASE)
   CF=0.902, PV=5.33844E-4, TP=27, FP=1, Pos=39, Neg=19
Pr 9. ((COL1A1 = -inf..5.706)) ==> (@Class = CONTROL)
   CF=0.884, PV=9.34323E-6, TP=14, FP=2, Pos=19, Neg=39
Nr 17. ((COL5A2 = -inf..5.552)) ==> (@Class = CONTROL)
   CF=0.938, PV=2.26509E-7, TP=14, FP=0, Pos=19, Neg=39

=== GEO4290 GEO4412 GEO16011 ===
Pr 2. ((COL6A3 = 9.672..inf)) ==> (@Class = CASE)
   CF=0.972, PV=2.80074E-3, TP=34, FP=0, Pos=81, Neg=19
Pr 20. ((COL1A1 = -inf..9.404)) ==> (@Class = CONTROL)
   CF=0.86, PV=6.47036E-6, TP=16, FP=4, Pos=26, Neg=59
Nr 28. ((COL6A1 = -inf..6.448)) ==> (@Class = CONTROL)
   CF=0.86, PV=1.08795E-6, TP=7, FP=3, Pos=16, Neg=159

=== GEO16011 ===
Pr 1. ((VEGFA = 9.367..inf)) ==> (@Class = CASE)
   CF=0.992, PV=7.47022E-5, TP=116, FP=0, Pos=159, Neg=16
Pr 3. ((VEGFA = -inf..7.475)) ==> (@Class = CONTROL)
   CF=0.877, PV=5.02443E-8, TP=9, FP=4, Pos=16, Neg=159

=== GEO4271 GEO1993 GEO4290 GEO16011 ===
Pr 7. ((VEGFA = -inf..8.676)) ==> (@Class = CONTROL)
```

```
   CF=0.904, PV=5.15453E-6, TP=13, FP=1, Pos=19, Neg=39
Pr 3. ((VEGFA = 10.677..inf)) ==> (@Class = CASE)
   CF=0.922, PV=7.3132E-5, TP=48, FP=1, Pos=76, Neg=24
Pr 4. ((VEGFA = 12.671..inf)) ==> (@Class = CASE)
   CF=0.911, PV=4.92388E-4, TP=53, FP=1, Pos=81, Neg=19
Nr 17. ((LDHA = 12.648..inf)) ==> (@Class = CASE)
   CF=0.98, PV=6.10817E-6, TP=49, FP=0, Pos=76, Neg=24
Pr 10. ((LDHA = -inf..10.110)) ==> (@Class = CONTROL)
   CF=0.884, PV=9.34323E-6, TP=14, FP=2, Pos=19, Neg=39
Nr 10. ((LDHA = 12.670..inf)) ==> (@Class = CASE)
   CF=0.991, PV=1.42534E-4, TP=109, FP=0, Pos=159, Neg=16

=== GEO4412 GEO4290 ===
Pr 6. ((IL8 = -inf..6.806)) ==> (@Class = CONTROL)
   CF=0.866, PV=3.01787E-5, TP=7, FP=1, Pos=19, Neg=81
Pr 19. ((IL8 = 11.240..inf)) ==> (@Class = CASE)
   CF=0.88, PV=4.27811E-3, TP=23, FP=1, Pos=59, Neg=26

=== GEO16011 GEO1993 GEO4271 GEO4290 ===
Pr 4. ((IGFBP2 = -inf..6.238)) ==> (@Class = CONTROL)
   CF=0.923, PV=5.8798E-6, TP=11, FP=0, Pos=19, Neg=39
Pr 2. ((IGFBP2 = 11.040..inf)) ==> (@Class = CASE)
   CF=0.99, PV=2.71578E-4, TP=102, FP=0, Pos=159, Neg=16
Pr 13. ((IGFBP2 = -inf..7.401)) ==> (@Class = CONTROL)
   CF=0.85, PV=5.71568E-5, TP=10, FP=3, Pos=24, Neg=76
Pr 9. ((IGFBP4 = -inf..7.710)) ==> (@Class = CONTROL)
   CF=0.824, PV=3.14646E-4, TP=7, FP=3, Pos=19, Neg=81

=== GEO4412 ===
Nr 32. ((ALDH5A1 = 12.977..inf)) ==> (@Class = CONTROL)
```

```
   CF=0.929, PV=8.14298E-7, TP=12, FP=0, Pos=26, Neg=59
Nr 35. ((ALDH6A1 = 13.840..inf)) ==> (@Class = CONTROL)
   CF=0.917, PV=8.1329E-6, TP=10, FP=0, Pos=26, Neg=59

=== GEO1993 GEO16011 GEO4412 ===
Pr 6. ((SERPINH1 = -inf..7.591)) ==> (@Class = CONTROL)
   CF=0.91, PV=1.84265E-6, TP=14, FP=1, Pos=19, Neg=39
Nr 24. ((SERPINH1 = 8.732..inf)) ==> (@Class = CASE)
   CF=0.905, PV=1.87403E-3, TP=103, FP=1, Pos=159, Neg=16
Pr 24. ((SERPINA3 = -inf..12.788)) ==> (@Class = CONTROL)
   CF=0.833, PV=4.85168E-5, TP=15, FP=5, Pos=26, Neg=59

=== GEO16011 GEO4412 ===
Pr 9. ((MELK = 11.481..inf)) ==> (@Class = CASE)
   CF=0.944, PV=7.88746E-3, TP=16, FP=0, Pos=59, Neg=26
Nr 30. ((MELK = -inf..7.069)) ==> (@Class = CONTROL)
   CF=0.851, PV=2.71454E-7, TP=11, FP=11, Pos=16, Neg=159
```

D.2.2 BREAST CANCER

```
=== GEO15852 ===
Pr 3. ((KRT18 = 7.214..inf)) ==> (@Class = CASE)
   CF=0.912, PV=2.20972E-7, TP=30, FP=2, Pos=43, Neg=43
Pr 6. ((KRT19 = -inf..5.649)) ==> (@Class = CONTROL)
   CF=0.895, PV=1.92648E-7, TP=33, FP=3, Pos=43, Neg=43
```

D.2.3 COLON CANCER

```
=== GEO20916 GEO24514 ===
Pr 1. ((MMP7 = 6.187..inf)) ==> (@Class = CASE)
```

```
   CF=0.974, PV=1.23811E-10, TP=36, FP=0, Pos=36, Neg=34
Pr 2. ((MMP7 = -inf..6.187)) ==> (@Class = CONTROL)
   CF=0.972, PV=6.25084E-11, TP=34, FP=0, Pos=34, Neg=36
Nr 3. ((CDH3 = 6.355..inf)) ==> (@Class = CASE)
   CF=0.974, PV=1.23811E-10, TP=36, FP=0, Pos=36, Neg=34
Nr 4. ((CDH3 = -inf..6.355)) ==> (@Class = CONTROL)
   CF=0.972, PV=6.25084E-11, TP=34, FP=0, Pos=34, Neg=36
Pr 2. ((MMP12 = 7.695..inf)) ==> (@Class = CASE)
   CF=0.968, PV=8.42805E-5, TP=29, FP=0, Pos=34, Neg=15

=== GEO23878 ===
Nr 3. ((ABCA8 = -inf..7.290)) ==> (@Class = CASE)
   CF=0.973, PV=3.893E-8, TP=35, FP=0, Pos=35, Neg=24
Nr 4. ((ABCA8 = 7.290..inf)) ==> (@Class = CONTROL)
   CF=0.962, PV=8.94963E-10, TP=24, FP=0, Pos=24, Neg=35
Pr 1. ((CXCL12 = -inf..8.567)) ==> (@Class = CASE)
   CF=0.97, PV=1.91249E-7, TP=31, FP=0, Pos=31, Neg=22
Pr 2. ((CXCL12 = 8.567..inf)) ==> (@Class = CONTROL)
   CF=0.958, PV=8.29696E-9, TP=22, FP=0, Pos=22, Neg=31
Pr 1. ((LPAR1 = -inf..7.721)) ==> (@Class = CASE)
   CF=0.971, PV=3.19315E-7, TP=32, FP=0, Pos=32, Neg=21
Pr 2. ((LPAR1 = 7.721..inf)) ==> (@Class = CONTROL)
   CF=0.957, PV=7.06764E-9, TP=21, FP=0, Pos=21, Neg=32

=== GEO24514 ===
Pr 4. ((MCM2 = -inf..9.183)) ==> (@Class = CONTROL)
   CF=0.938, PV=8.18121E-8, TP=14, FP=0, Pos=15, Neg=34
Pr 5. ((MCM2 = 9.183..inf)) ==> (@Class = CASE)
   CF=0.915, PV=1.22339E-4, TP=34, FP=1, Pos=34, Neg=15
Pr 3. ((PMAIP1 = -inf..6.326)) ==> (@Class = CONTROL)
```

```
   CF=0.929, PV=5.17967E-7, TP=12, FP=0, Pos=13, Neg=31
Pr 4. ((PMAIP1 = 6.326..inf)) ==> (@Class = CASE)
   CF=0.904, PV=5.10453E-4, TP=31, FP=1, Pos=31, Neg=13
Nr 5. ((TEX10 = 8.359..inf)) ==> (@Class = CASE)
   CF=0.97, PV=6.71667E-5, TP=31, FP=0, Pos=31, Neg=13
Nr 6. ((TEX10 = -inf..8.359)) ==> (@Class = CONTROL)
   CF=0.933, PV=1.57808E-7, TP=13, FP=0, Pos=13, Neg=31

=== GEO9348 ===
Pr 1. ((INHBA = 5.114..inf)) ==> (@Class = CASE)
   CF=0.985, PV=1.75372E-4, TP=63, FP=0, Pos=63, Neg=11
Pr 2. ((INHBA = -inf..5.114)) ==> (@Class = CONTROL)
   CF=0.923, PV=4.19698E-10, TP=11, FP=0, Pos=11, Neg=63

=== GEO10715 ===
Pr 3. ((LILRB4 = 6.147..inf)) ==> (@Class = CASE)
   CF=0.863, PV=8.54581E-3, TP=16, FP=1, Pos=17, Neg=10
Pr 4. ((LILRB4 = -inf..6.147)) ==> (@Class = CONTROL)
   CF=0.863, PV=9.60573E-4, TP=9, FP=1, Pos=10, Neg=17
```

D.2.4 LUNG CANCER

```
=== GEO10072 ===
Pr 1. ((EDNRB = -inf..7.480)) ==> (@Class = CASE)
   CF=0.983, PV=1.44329E-15, TP=58, FP=0, Pos=58, Neg=49
Pr 2. ((EDNRB = 7.480..inf)) ==> (@Class = CONTROL)
   CF=0.98, PV=1.11022E-16, TP=49, FP=0, Pos=49, Neg=58
Nr 3. ((EDNRB = -inf..7.480)) ==> (@Class = CASE)
   CF=0.981, PV=5.37348E-14, TP=52, FP=0, Pos=52, Neg=44
Pr 1. ((PECAM1 = -inf..10.741)) ==> (@Class = CASE)
```

```
   CF=0.981, PV=5.37348E-14, TP=52, FP=0, Pos=52, Neg=44
Pr 2. ((PECAM1 = 10.741..inf)) ==> (@Class = CONTROL)
   CF=0.978, PV=3.77476E-15, TP=44, FP=0, Pos=44, Neg=52

=== GEO18842 GEO19188 ===
Pr 1. ((CENPE = 5.343..inf)) ==> (@Class = CASE)
   CF=0.979, PV=7.9603E-14, TP=46, FP=0, Pos=46, Neg=45
Pr 2. ((CENPE = -inf..5.343)) ==> (@Class = CONTROL)
Pr 1. ((PLK4 = 5.412..inf)) ==> (@Class = CASE)
   CF=0.977, PV=2.05735E-12, TP=42, FP=0, Pos=42, Neg=40
Nr 3. ((AQP4 = -inf..10.455)) ==> (@Class = CASE)
   CF=0.977, PV=1.45239E-12, TP=41, FP=0, Pos=41, Neg=41
Nr 4. ((AQP4 = 10.455..inf)) ==> (@Class = CONTROL)
   CF=0.977, PV=1.45239E-12, TP=41, FP=0, Pos=41, Neg=41
Pr 1. ((AQP1 = -inf..11.015)) ==> (@Class = CASE)
   CF=0.988, PV=0E0, TP=84, FP=0, Pos=91, Neg=65
Pr 3. ((AQP1 = 11.015..12.226)) ==> (@Class = CONTROL)
   CF=0.974, PV=0E0, TP=63, FP=1, Pos=65, Neg=91
Nr 4. ((PLK4 = -inf..5.412)) ==> (@Class = CONTROL)
   CF=0.976, PV=1.04927E-12, TP=40, FP=0, Pos=40, Neg=42

=== GEO19804 ===
Pr 14. ((AGER = -inf..9.445)) ==> (@Class = CASE)
   CF=0.952, PV=3.55271E-15, TP=58, FP=2, Pos=60, Neg=60
Pr 15. ((AGER = 9.445..inf)) ==> (@Class = CONTROL)
   CF=0.952, PV=3.55271E-15, TP=58, FP=2, Pos=60, Neg=60
Pr 4. ((CDH3 = -inf..6.161)) ==> (@Class = CONTROL)
   CF=0.976, PV=5.68434E-12, TP=39, FP=0, Pos=60, Neg=60
Pr 8. ((CDH3 = 7.506..inf)) ==> (@Class = CASE)
   CF=0.975, PV=1.12412E-11, TP=38, FP=0, Pos=60, Neg=60
```

```
=== GEO19804 GEO18842 ===
Pr 9. ((CCNB1 = 6.633..inf)) ==> (@Class = CASE)
   CF=0.974, PV=2.22247E-11, TP=37, FP=0, Pos=60, Neg=60
Nr 4. ((CCNB1 = -inf..6.983)) ==> (@Class = CONTROL)
   CF=0.979, PV=5.70655E-14, TP=45, FP=0, Pos=45, Neg=46
```

D.2.5 PROSTATE CANCER

```
 === GEO46602 GEO17951 ===
Pr 1. ((HPN = 7.963..inf)) ==> (@Class = CASE)
   CF=0.974, PV=2.71643E-5, TP=36, FP=0, Pos=36, Neg=14
Pr 2. ((HPN = -inf..7.963)) ==> (@Class = CONTROL)
   CF=0.938, PV=2.1384E-8, TP=14, FP=0, Pos=14, Neg=36
Pr 3. ((HPN = 7.807..inf)) ==> (@Class = CASE)
   CF=0.968, PV=1.11022E-16, TP=59, FP=1, Pos=68, Neg=69

=== GEO82188 GEO17951 ===
Pr 1. ((HPN = -inf..7.605)) ==> (@Class = CONTROL)
   CF=0.979, PV=3.5949E-13, TP=46, FP=0, Pos=71, Neg=65
Pr 8. ((HPN = 8.949..inf)) ==> (@Class = CASE)
   CF=0.949, PV=1.77636E-14, TP=52, FP=2, Pos=65, Neg=71
Pr 7. ((CLU = 11.518..inf)) ==> (@Class = CONTROL)
   CF=0.958, PV=3.20943E-12, TP=47, FP=1, Pos=71, Neg=65
Pr 10. ((CLU = -inf..10.121)) ==> (@Class = CASE)
   CF=0.936, PV=6.3712E-8, TP=27, FP=1, Pos=65, Neg=71
Pr 11. ((GDF15 = -inf..9.977)) ==> (@Class = CONTROL)
   CF=0.931, PV=6.55994E-10, TP=42, FP=2, Pos=71, Neg=65
Pr 15. ((GDF15 = 11.976..inf)) ==> (@Class = CASE)
```

```
   CF=0.908, PV=6.71046E-9, TP=36, FP=3, Pos=65, Neg=71
Pr 14. ((TRPM4 = -inf..8.995)) ==> (@Class = CONTROL)
   CF=0.91, PV=4.94493E-13, TP=64, FP=5, Pos=71, Neg=65
Pr 17. ((TRPM4 = 8.995..inf)) ==> (@Class = CASE)
   CF=0.892, PV=9.43023E-13, TP=60, FP=7, Pos=65, Neg=71
Pr 3. ((EPCAM = 10.407..inf)) ==> (@Class = CASE)
   CF=0.966, PV=4.89446E-9, TP=27, FP=0, Pos=58, Neg=64
Pr 4. ((EPCAM = -inf..8.545)) ==> (@Class = CONTROL)
   CF=0.964, PV=1.41931E-7, TP=26, FP=0, Pos=64, Neg=58

=== GEO17951 GEO32448 ===
Nr 20. ((FGF2 = 8.546..inf)) ==> (@Class = CONTROL)
   CF=0.97, PV=1.66195E-9, TP=31, FP=0, Pos=69, Neg=68
Pr 7. ((FGFR2 = 7.425..inf)) ==> (@Class = CONTROL)
   CF=0.903, PV=1.41289E-6, TP=27, FP=2, Pos=40, Neg=40

=== GEO82188 GEO32448 ===
Pr 6. ((ID4 = -inf..7.554)) ==> (@Class = CASE)
   CF=0.96, PV=1.01793E-7, TP=23, FP=0, Pos=65, Neg=71
Pr 6. ((ID4 = 9.140..inf)) ==> (@Class = CONTROL)
   CF=0.913, PV=1.89246E-5, TP=20, FP=1, Pos=40, Neg=40

=== GEO32448 GEO4660 ===
Nr 19. ((CYP3A5 = 6.448..inf)) ==> (@Class = CONTROL)
   CF=0.938, PV=5.89155E-8, TP=29, FP=1, Pos=36, Neg=36
Nr 3. ((CYP3A5 = -inf..6.786)) ==> (@Class = CASE)
   CF=0.971, PV=6.46417E-5, TP=32, FP=0, Pos=32, Neg=13
```

APPENDIX E

MS-TRL - SUPPLEMENTARY

E.1 CLASSIFICATION PERFORMANCE

Table 67: Classification performance of MS-TRL Combo, two source datasets, on brain cancer set

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**	**Ab**
GEO4271_GEO4290	GEO16011	96.855	6.250	51.553	88.571	0.000
GEO4271_GEO4412	GEO16011	96.855	6.250	51.553	88.571	0.000
GEO1993_GEO4271	GEO16011	96.855	12.500	54.678	89.143	0.000
GEO4290_GEO4412	GEO16011	98.734	0.000	49.367	89.143	1.143
GEO1993_GEO4412	GEO16011	98.113	12.500	55.307	90.286	0.000
GEO1993_GEO4290	GEO16011	98.742	12.500	55.621	90.857	0.000
GEO4271_GEO4412	GEO1993	87.179	73.684	80.432	82.759	0.000
GEO4290_GEO4412	GEO1993	94.872	63.158	79.015	84.483	0.000
GEO4271_GEO4290	GEO1993	92.308	73.684	82.996	86.207	0.000
GEO4271_GEO16011	GEO1993	92.308	73.684	82.996	86.207	0.000
GEO16011_GEO4290	GEO1993	97.436	68.421	82.928	87.931	0.000
GEO16011_GEO4412	GEO1993	97.436	68.421	82.928	87.931	0.000
GEO4290_GEO4412	GEO4271	97.368	20.833	59.101	79.000	0.000
GEO1993_GEO4412	GEO4271	98.684	20.833	59.759	80.000	0.000
GEO16011_GEO4412	GEO4271	100.000	20.833	60.417	81.000	0.000
GEO1993_GEO4290	GEO4271	98.684	25.000	61.842	81.000	0.000
GEO16011_GEO4290	GEO4271	100.000	20.833	60.417	81.000	0.000
GEO1993_GEO16011	GEO4271	100.000	20.833	60.417	81.000	0.000
GEO4271_GEO4412	GEO4290	93.827	36.842	65.335	83.000	0.000
GEO16011_GEO4412	GEO4290	95.062	31.579	63.320	83.000	0.000
GEO4271_GEO16011	GEO4290	96.296	31.579	63.938	84.000	0.000
GEO1993_GEO4271	GEO4290	96.296	31.579	63.938	84.000	0.000
GEO1993_GEO4412	GEO4290	95.062	36.842	65.952	84.000	0.000
GEO1993_GEO16011	GEO4290	98.765	31.579	65.172	86.000	0.000
GEO16011_GEO4290	GEO4412	96.610	50.000	73.305	82.353	0.000
GEO4271_GEO16011	GEO4412	94.915	57.692	76.304	83.529	0.000
GEO1993_GEO4271	GEO4412	93.220	61.538	77.379	83.529	0.000
GEO4271_GEO4290	GEO4412	96.610	57.692	77.151	84.706	0.000
GEO1993_GEO16011	GEO4412	91.525	69.231	80.378	84.706	0.000
GEO1993_GEO4290	GEO4412	96.610	65.385	80.997	87.059	0.000

Table 68: Classification performance of MS-TRL Combo, three source datasets, on brain cancer set

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**
GEO4271_GEO4290_GEO4412	GEO16011	96.855	6.250	51.553	88.571
GEO1993_GEO4271_GEO4290	GEO16011	96.855	12.500	54.678	89.143
GEO1993_GEO4271_GEO4412	GEO16011	96.855	12.500	54.678	89.143
GEO1993_GEO4290_GEO4412	GEO16011	98.742	12.500	55.621	90.857
GEO4271_GEO4290_GEO4412	GEO1993	89.744	68.421	79.082	82.759
GEO4271_GEO16011_GEO4412	GEO1993	92.308	68.421	80.364	84.483
GEO16011_GEO4290_GEO4412	GEO1993	94.872	68.421	81.646	86.207
GEO4271_GEO16011_GEO4290	GEO1993	94.872	68.421	81.646	86.207
GEO16011_GEO4290_GEO4412	GEO4271	98.684	20.833	59.759	80.000
GEO1993_GEO4290_GEO4412	GEO4271	97.368	25.000	61.184	80.000
GEO1993_GEO16011_GEO4412	GEO4271	100.000	20.833	60.417	81.000
GEO1993_GEO16011_GEO4290	GEO4271	98.684	25.000	61.842	81.000
GEO4271_GEO16011_GEO4412	GEO4290	93.827	26.316	63.000	81.000
GEO1993_GEO4271_GEO4412	GEO4290	93.827	36.842	65.335	83.000
GEO1993_GEO16011_GEO4412	GEO4290	95.062	36.842	65.952	84.000
GEO1993_GEO4271_GEO16011	GEO4290	96.296	31.579	63.938	84.000
GEO4271_GEO16011_GEO4290	GEO4412	98.305	53.846	76.076	84.706
GEO1993_GEO4271_GEO16011	GEO4412	94.915	61.538	78.227	84.706
GEO1993_GEO16011_GEO4290	GEO4412	96.610	61.538	79.074	85.882
GEO1993_GEO4271_GEO4290	GEO4412	96.610	61.538	79.074	85.882

Table 69: Classification performance of MS-TRL Combo, using two source datasets within the mixed cancer set

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**	**Ab**
GEO7904_GEO7670	GEO4412	93.22	50.00	71.61	80.00	0.00
GEO9348_GEO7670	GEO4412	93.10	53.85	73.48	80.00	1.18
GEO7904_GEO9348	GEO4412	96.55	50.00	73.28	81.18	1.18
GEO6956_GEO9348	GEO4412	94.92	53.85	74.38	82.35	0.00
GEO6956_GEO7670	GEO4412	94.92	53.85	74.38	82.35	0.00
GEO6956_GEO7904	GEO4412	98.31	50.00	74.15	83.53	0.00
GEO9348_GEO7670	GEO6956	94.12	55.00	74.56	84.27	1.12
GEO7670_GEO4412	GEO6956	95.59	55.00	75.29	85.39	1.12
GEO9348_GEO4412	GEO6956	95.59	55.00	75.29	85.39	1.12
GEO7904_GEO4412	GEO6956	97.10	55.00	76.05	87.64	0.00
GEO7904_GEO7670	GEO6956	98.55	55.00	76.78	88.76	0.00
GEO7904_GEO9348	GEO6956	98.55	55.00	76.78	88.76	0.00
GEO6956_GEO7904	GEO7670	94.87	81.48	88.18	89.39	0.00
GEO6956_GEO9348	GEO7670	94.87	81.48	88.18	89.39	0.00
GEO9348_GEO4412	GEO7670	94.87	85.19	90.03	90.91	0.00
GEO7904_GEO9348	GEO7670	94.87	85.19	90.03	90.91	0.00
GEO6956_GEO4412	GEO7670	94.87	85.19	90.03	90.91	0.00
GEO7904_GEO4412	GEO7670	97.44	85.19	91.31	92.42	0.00
GEO6956_GEO9348	GEO7904	93.02	31.58	62.30	74.19	0.00
GEO6956_GEO4412	GEO7904	95.35	31.58	63.46	75.81	0.00
GEO9348_GEO7670	GEO7904	95.35	31.58	63.46	75.81	0.00
GEO9348_GEO4412	GEO7904	95.35	36.84	66.10	77.42	0.00
GEO6956_GEO7670	GEO7904	97.67	36.84	67.26	79.03	0.00
GEO7670_GEO4412	GEO7904	97.67	36.84	67.26	79.03	0.00
GEO7904_GEO7670	GEO9348	100.00	66.67	83.33	95.12	0.00
GEO6956_GEO4412	GEO9348	100.00	66.67	83.33	95.12	0.00
GEO7670_GEO4412	GEO9348	100.00	66.67	83.33	95.12	0.00
GEO6956_GEO7670	GEO9348	100.00	75.00	87.50	96.34	0.00
GEO7904_GEO4412	GEO9348	100.00	83.33	91.67	97.56	0.00
GEO6956_GEO7904	GEO9348	100.00	83.33	91.67	97.56	0.00

Table 70: Classification performance of MS-TRL Combo, using three source datasets within the mixed cancer set

Source	Target	SN	SP	BACC	AccAb	Ab
GEO6956_GEO7904_GEO9348	GEO4412	94.915	46.154	70.535	80.000	0.000
GEO7904_GEO9348_GEO7670	GEO4412	94.828	50.000	72.414	80.000	1.176
GEO6956_GEO9348_GEO7670	GEO4412	93.220	53.846	73.533	81.176	0.000
GEO6956_GEO7904_GEO7670	GEO4412	98.305	50.000	74.153	83.529	0.000
GEO9348_GEO7670_GEO4412	GEO6956	94.118	55.000	74.559	84.270	1.124
GEO7904_GEO7670_GEO4412	GEO6956	97.101	55.000	76.051	87.640	0.000
GEO7904_GEO9348_GEO4412	GEO6956	97.101	55.000	76.051	87.640	0.000
GEO7904_GEO9348_GEO7670	GEO6956	97.101	55.000	76.051	87.640	0.000
GEO6956_GEO7904_GEO9348	GEO7670	94.872	81.481	88.177	89.394	0.000
GEO6956_GEO7904_GEO4412	GEO7670	94.872	81.481	88.177	89.394	0.000
GEO6956_GEO9348_GEO4412	GEO7670	94.872	88.889	91.880	92.424	0.000
GEO7904_GEO9348_GEO4412	GEO7670	97.436	85.185	91.311	92.424	0.000
GEO6956_GEO9348_GEO4412	GEO7904	93.023	31.579	62.301	74.194	0.000
GEO9348_GEO7670_GEO4412	GEO7904	97.674	26.316	61.995	75.806	0.000
GEO6956_GEO9348_GEO7670	GEO7904	95.349	36.842	66.095	77.419	0.000
GEO6956_GEO7670_GEO4412	GEO7904	97.674	31.579	64.627	77.419	0.000
GEO7904_GEO7670_GEO4412	GEO9348	100.000	58.333	79.167	93.902	0.000
GEO6956_GEO7670_GEO4412	GEO9348	100.000	66.667	83.333	95.122	0.000
GEO6956_GEO7904_GEO7670	GEO9348	100.000	66.667	83.333	95.122	0.000
GEO6956_GEO7904_GEO4412	GEO9348	100.000	75.000	87.500	96.341	0.000

Table 71: Classification performance of MS-TRL OnlyPriors, using two source datasets within the brain cancer set

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**	**Ab**
GEO1993_GEO4290	GEO16011	96.599	50.000	73.299	85.143	8.000
GEO4290_GEO4412	GEO16011	98.561	20.000	59.281	78.857	17.714
GEO1993_GEO4412	GEO16011	95.425	50.000	72.712	87.429	4.571
GEO4271_GEO4412	GEO16011	89.103	50.000	69.551	82.286	5.143
GEO1993_GEO4271	GEO16011	91.613	57.143	74.378	85.714	3.429
GEO4271_GEO4290	GEO16011	90.850	54.545	72.698	82.857	6.286
GEO4271_GEO4412	GEO1993	87.179	47.368	67.274	74.138	0.000
GEO4271_GEO4290	GEO1993	94.737	57.895	76.316	81.034	1.724
GEO4290_GEO4412	GEO1993	97.368	44.444	70.906	77.586	3.448
GEO4271_GEO16011	GEO1993	91.429	64.706	78.067	74.138	10.345
GEO16011_GEO4290	GEO1993	100.000	50.000	75.000	77.586	6.897
GEO16011_GEO4412	GEO1993	94.737	61.111	77.924	81.034	3.448
GEO1993_GEO16011	GEO4271	94.667	54.545	74.606	83.000	3.000
GEO1993_GEO4412	GEO4271	93.333	54.545	73.939	82.000	3.000
GEO16011_GEO4412	GEO4271	93.421	50.000	71.711	82.000	2.000
GEO1993_GEO4290	GEO4271	93.421	43.478	68.450	81.000	1.000
GEO16011_GEO4290	GEO4271	96.053	52.174	74.113	85.000	1.000
GEO4290_GEO4412	GEO4271	93.421	54.545	73.983	83.000	2.000
GEO4271_GEO16011	GEO4290	88.462	38.889	63.675	76.000	4.000
GEO4271_GEO4412	GEO4290	88.462	52.632	70.547	79.000	3.000
GEO1993_GEO4271	GEO4290	86.420	52.632	69.526	80.000	0.000
GEO1993_GEO4412	GEO4290	85.897	57.895	71.896	78.000	3.000
GEO1993_GEO16011	GEO4290	85.000	57.895	71.447	79.000	1.000
GEO16011_GEO4412	GEO4290	87.342	47.368	67.355	78.000	2.000
GEO1993_GEO16011	GEO4412	98.246	56.522	77.384	81.176	5.882
GEO1993_GEO4271	GEO4412	91.228	66.667	78.947	80.000	4.706
GEO1993_GEO4290	GEO4412	87.500	60.000	73.750	75.294	4.706
GEO4271_GEO4290	GEO4412	91.525	46.154	68.840	77.647	0.000
GEO16011_GEO4290	GEO4412	96.491	41.667	69.079	76.471	4.706
GEO4271_GEO16011	GEO4412	98.214	43.478	70.846	76.471	7.059

Table 72: Classification performance of MS-TRL OnlyPriors, using three source datasets within the brain cancer set

Source	Target	SN	SP	BACC	AccAb	Ab
GEO4271_GEO4290_GEO4412	GEO16011	90.968	44.444	67.706	82.857	6.286
GEO1993_GEO4271_GEO4290	GEO16011	92.258	57.143	74.700	86.286	3.429
GEO1993_GEO4271_GEO4412	GEO16011	92.949	57.143	75.046	87.429	2.857
GEO1993_GEO4290_GEO4412	GEO16011	96.732	50.000	73.366	88.571	4.571
GEO16011_GEO4290_GEO4412	GEO1993	97.436	43.750	70.593	77.586	5.172
GEO4271_GEO16011_GEO4290	GEO1993	94.595	57.895	76.245	79.310	3.448
GEO4271_GEO4290_GEO4412	GEO1993	94.737	52.632	73.684	79.310	1.724
GEO4271_GEO16011_GEO4412	GEO1993	92.105	57.895	75.000	79.310	1.724
GEO1993_GEO16011_GEO4412	GEO4271	94.667	43.478	69.072	81.000	2.000
GEO16011_GEO4290_GEO4412	GEO4271	94.737	40.909	67.823	81.000	2.000
GEO1993_GEO4290_GEO4412	GEO4271	93.333	43.478	68.406	80.000	2.000
GEO1993_GEO16011_GEO4290	GEO4271	96.053	39.130	67.592	82.000	1.000
GEO1993_GEO16011_GEO4412	GEO4290	88.750	57.895	73.322	82.000	1.000
GEO1993_GEO4271_GEO4412	GEO4290	86.250	52.632	69.441	79.000	1.000
GEO1993_GEO4271_GEO16011	GEO4290	87.500	52.632	70.066	80.000	1.000
GEO4271_GEO16011_GEO4412	GEO4290	91.139	47.368	69.254	81.000	2.000
GEO4271_GEO16011_GEO4290	GEO4412	98.214	38.462	68.338	76.471	3.529
GEO1993_GEO4271_GEO16011	GEO4412	92.982	60.870	76.926	78.824	5.882
GEO1993_GEO16011_GEO4290	GEO4412	94.737	48.000	71.368	77.647	3.529
GEO1993_GEO4271_GEO4290	GEO4412	90.909	52.000	71.455	74.118	5.882

Table 73: Classification performance of MS-TRL OnlyPriors, using two source datasets within the mixed cancer set

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**	**Ab**
GEO7904_GEO9348	GEO4412	84.906	37.500	61.203	60.000	18.824
GEO6956_GEO7904	GEO4412	91.379	52.381	71.880	75.294	7.059
GEO7904_GEO7670	GEO4412	91.071	50.000	70.536	71.765	10.588
GEO6956_GEO9348	GEO4412	83.636	68.421	76.029	69.412	12.941
GEO6956_GEO7670	GEO4412	94.828	52.381	73.604	77.647	7.059
GEO9348_GEO7670	GEO4412	90.909	54.167	72.538	74.118	7.059
GEO7904_GEO4412	GEO6956	96.296	55.556	75.926	69.663	19.101
GEO7904_GEO7670	GEO6956	91.667	55.000	73.333	74.157	10.112
GEO9348_GEO7670	GEO6956	61.765	87.500	74.632	39.326	43.820
GEO7904_GEO9348	GEO6956	95.455	50.000	72.727	56.180	32.584
GEO7670_GEO4412	GEO6956	88.462	55.556	72.009	62.921	21.348
GEO9348_GEO4412	GEO6956	83.721	61.111	72.416	52.809	31.461
GEO9348_GEO4412	GEO7670	91.667	80.769	86.218	81.818	6.061
GEO7904_GEO9348	GEO7670	92.308	92.000	92.154	89.394	3.030
GEO6956_GEO7904	GEO7670	97.436	83.333	90.385	87.879	4.545
GEO6956_GEO9348	GEO7670	94.872	80.769	87.821	87.879	1.515
GEO6956_GEO4412	GEO7670	97.368	80.000	88.684	86.364	4.545
GEO7904_GEO4412	GEO7670	92.105	84.000	88.053	84.848	4.545
GEO6956_GEO4412	GEO7904	94.595	33.333	63.964	64.516	16.129
GEO6956_GEO7670	GEO7904	90.476	40.000	65.238	70.968	8.065
GEO9348_GEO7670	GEO7904	90.698	35.714	63.206	70.968	8.065
GEO7670_GEO4412	GEO7904	95.349	31.250	63.299	74.194	4.839
GEO9348_GEO4412	GEO7904	92.308	40.000	66.154	67.742	12.903
GEO6956_GEO9348	GEO7904	84.211	33.333	58.772	58.065	19.355
GEO7904_GEO4412	GEO9348	100.000	55.556	77.778	91.463	3.659
GEO7904_GEO7670	GEO9348	100.000	50.000	75.000	92.683	0.000
GEO6956_GEO4412	GEO9348	100.000	50.000	75.000	92.683	0.000
GEO6956_GEO7670	GEO9348	100.000	58.333	79.167	93.902	0.000
GEO7670_GEO4412	GEO9348	100.000	33.333	66.667	90.244	0.000
GEO6956_GEO7904	GEO9348	98.571	83.333	90.952	96.341	0.000

Table 74: Classification performance of MS-TRL OnlyPriors, using three source datasets within the mixed cancer set

Source	Target	SN	SP	BACC	AccAb	Ab
GEO7904_GEO9348_GEO7670	GEO4412	91.228	50.000	70.614	75.294	4.706
GEO6956_GEO7904_GEO9348	GEO4412	91.379	57.143	74.261	76.471	7.059
GEO6956_GEO9348_GEO7670	GEO4412	91.379	52.174	71.777	76.471	4.706
GEO6956_GEO7904_GEO7670	GEO4412	92.982	52.174	72.578	76.471	5.882
GEO7904_GEO9348_GEO7670	GEO6956	90.625	55.000	72.812	77.528	5.618
GEO9348_GEO7670_GEO4412	GEO6956	87.500	60.000	73.750	68.539	14.607
GEO7904_GEO7670_GEO4412	GEO6956	92.063	50.000	71.032	76.404	6.742
GEO7904_GEO9348_GEO4412	GEO6956	94.737	57.895	76.316	73.034	14.607
GEO6956_GEO9348_GEO4412	GEO7670	97.368	80.769	89.069	87.879	3.030
GEO6956_GEO7904_GEO9348	GEO7670	94.872	76.000	85.436	84.848	3.030
GEO6956_GEO7904_GEO4412	GEO7670	94.872	83.333	89.103	86.364	4.545
GEO7904_GEO9348_GEO4412	GEO7670	94.872	88.462	91.667	90.909	1.515
GEO9348_GEO7670_GEO4412	GEO7904	95.349	25.000	60.174	72.581	4.839
GEO6956_GEO9348_GEO7670	GEO7904	90.698	40.000	65.349	72.581	6.452
GEO6956_GEO7670_GEO4412	GEO7904	93.023	37.500	65.262	74.194	4.839
GEO6956_GEO9348_GEO4412	GEO7904	88.095	37.500	62.798	69.355	6.452
GEO6956_GEO7904_GEO7670	GEO9348	100.000	50.000	75.000	92.683	0.000
GEO6956_GEO7670_GEO4412	GEO9348	100.000	50.000	75.000	92.683	0.000
GEO6956_GEO7904_GEO4412	GEO9348	100.000	66.667	83.333	95.122	0.000
GEO7904_GEO7670_GEO4412	GEO9348	100.000	41.667	70.833	91.463	0.000

Table 75: The average AccAb (%) per number of sources for MS-TRL-Combo. For number of sources, None ≡ RL, and One ≡ TRL.

		Number of sources				
Disease	**Target**	**None**	**One**	**Two**	**Three**	**Four**
Brain	GEO16011	89.143	89.286	89.429	89.429	89.143
Brain	GEO1993	86.207	86.638	85.920	84.914	84.483
Brain	GEO4271	78.000	80.000	80.500	80.500	80.000
Brain	GEO4290	82.000	83.750	84.000	83.000	83.000
Brain	GEO4412	81.176	82.059	84.313	85.294	84.706
Breast	GEO10780	86.486	87.162	87.658	87.973	88.108
Breast	GEO15852	84.884	85.466	85.078	84.884	86.047
Breast	GEO29431	93.939	95.076	95.758	95.833	95.455
Breast	GEO42568	95.868	96.281	95.868	95.248	95.041
Breast	GEO7904	77.419	77.822	77.150	77.016	77.419
Colon	GEO10715	60.000	67.500	68.888	69.167	70.000
Colon	GEO20916	97.143	97.857	98.333	97.857	97.143
Colon	GEO23878	94.915	95.763	96.610	96.610	98.305
Colon	GEO24514	93.878	96.428	96.599	97.449	97.959
Colon	GEO9348	95.122	97.561	99.797	100.000	100.000
Lung	GEO10072	96.262	96.729	95.794	95.327	95.327
Lung	GEO18842	91.209	97.527	98.166	99.176	100.000
Lung	GEO19188	94.872	95.513	95.299	95.193	94.872
Lung	GEO19804	94.167	94.792	94.445	94.584	95.000
Lung	GEO7670	87.879	93.940	94.192	93.939	93.939
Mix	GEO4412	81.176	82.059	81.569	81.176	82.353
Mix	GEO6956	87.640	86.236	86.704	86.798	86.517
Mix	GEO7904	87.879	89.053	90.657	90.909	89.394
Mix	GEO7670	77.419	78.629	76.882	76.210	77.419
Mix	GEO9348	95.122	96.951	96.138	95.122	95.122
Prostate	GEO17951	83.212	85.949	86.496	86.314	86.131
Prostate	GEO32448	88.750	90.313	91.666	92.813	93.750
Prostate	GEO46602	92.000	94.000	94.000	94.000	94.000
Prostate	GEO6956	87.640	86.236	86.142	83.989	84.270
Prostate	GEO82188	87.500	90.809	90.319	89.706	89.706

Table 76: The average AccAb (%) per number of sources for MS-TRL-OnlyPriors. For number of sources, None ≡ RL, and One ≡ TRL

		Number of sources				
Disease	**Target**	**None**	**One**	**Two**	**Three**	**Four**
Brain	GEO16011	89.143	78.286	83.714	86.286	87.429
Brain	GEO1993	86.207	74.569	77.586	78.879	81.034
Brain	GEO4271	78.000	80.250	82.666	81.000	82.000
Brain	GEO4290	82.000	74.250	78.333	80.500	79.000
Brain	GEO4412	81.176	73.824	77.843	76.765	76.471
Breast	GEO10780	86.486	31.622	55.766	69.595	76.216
Breast	GEO15852	84.884	63.663	83.721	87.210	86.047
Breast	GEO29431	93.939	82.576	93.939	94.697	95.455
Breast	GEO42568	95.868	84.504	95.592	95.455	95.041
Breast	GEO7904	77.419	39.516	57.796	64.516	69.355
Colon	GEO10715	60.000	35.000	59.445	72.500	80.000
Colon	GEO20916	97.143	74.999	90.429	96.040	95.714
Colon	GEO23878	94.915	73.305	90.678	94.915	96.610
Colon	GEO24514	93.878	78.565	89.116	96.939	97.959
Colon	GEO9348	95.122	87.195	98.577	99.085	100.000
Lung	GEO10072	96.262	85.981	92.679	92.991	93.458
Lung	GEO18842	91.209	92.857	96.520	95.879	94.505
Lung	GEO19188	94.872	90.705	93.590	93.750	94.872
Lung	GEO19804	94.167	82.500	89.861	92.500	93.333
Lung	GEO7670	87.879	90.909	93.182	93.939	93.939
Mix	GEO4412	81.176	52.647	71.373	76.177	76.471
Mix	GEO6956	87.640	32.023	59.176	73.876	78.652
Mix	GEO7904	87.879	71.970	86.364	87.500	84.848
Mix	GEO7670	77.419	48.791	67.742	72.178	72.581
Mix	GEO9348	95.122	93.902	92.886	92.988	92.683
Prostate	GEO17951	83.212	67.336	79.927	81.752	83.212
Prostate	GEO32448	88.750	65.313	74.792	80.000	81.250
Prostate	GEO46602	92.000	91.500	92.333	93.000	94.000
Prostate	GEO6956	87.640	58.708	71.910	76.405	77.528
Prostate	GEO82188	87.500	69.669	82.598	84.375	83.824

Table 77: The average rate (%) of abstentions per number of sources for MS-TRL-OnlyPriors

		Number of sources				
Disease	Target	None	One	Two	Three	Four
Brain	GEO16011	1.143	15.714	7.523	4.286	2.857
Brain	GEO1993	0.000	5.604	4.310	3.017	1.724
Brain	GEO4271	0.000	4.250	2.000	1.750	2.000
Brain	GEO4290	1.000	7.750	2.167	1.333	3.000
Brain	GEO4412	1.176	7.941	4.533	4.706	4.706
Breast	GEO10780	1.081	59.189	27.838	10.676	4.865
Breast	GEO15852	0.000	24.709	5.039	2.035	2.326
Breast	GEO29431	6.061	8.712	1.515	0.758	0.000
Breast	GEO42568	1.653	8.265	0.138	0.207	0.000
Breast	GEO7904	1.613	39.919	18.011	11.291	6.452
Colon	GEO10715	13.333	54.167	22.222	6.667	3.333
Colon	GEO20916	1.429	18.214	2.619	1.429	1.429
Colon	GEO23878	5.085	19.068	3.108	0.848	0.000
Colon	GEO24514	0.000	30.102	4.762	0.000	0.000
Colon	GEO9348	2.439	8.232	0.000	0.000	0.000
Lung	GEO10072	1.869	7.710	2.181	2.337	1.869
Lung	GEO18842	4.396	4.121	2.198	1.923	2.198
Lung	GEO19188	0.000	2.885	0.534	0.641	0.000
Lung	GEO19804	0.833	7.500	1.389	0.625	0.833
Lung	GEO7670	4.545	2.273	1.515	1.136	0.000
Mix	GEO4412	1.176	32.647	10.588	5.588	3.529
Mix	GEO6956	1.124	57.584	26.404	10.394	5.618
Mix	GEO7904	1.613	35.484	11.559	5.646	4.839
Mix	GEO7670	4.545	18.182	4.040	3.030	3.030
Mix	GEO9348	2.439	1.525	0.610	0.000	0.000
Prostate	GEO17951	3.650	15.694	2.312	1.095	2.190
Prostate	GEO32448	5.000	20.000	8.125	4.063	2.500
Prostate	GEO46602	4.000	1.500	1.000	1.000	0.000
Prostate	GEO6956	1.124	22.472	6.929	3.652	3.371
Prostate	GEO82188	2.206	16.177	3.677	2.757	2.941

Table 78: Accuracy of MS-TRL with two best sources

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**
GEO1993_GEO4290	GEO16011	98.742	12.500	55.621	90.857
GEO16011_GEO4412	GEO1993	97.436	68.421	82.928	87.931
GEO1993_GEO16011	GEO4271	100.000	20.833	60.417	81.000
GEO1993_GEO16011	GEO4290	98.765	31.579	65.172	86.000
GEO1993_GEO4290	GEO4412	96.610	65.385	80.997	87.059
GEO7904_GEO42568	GEO10780	50.000	100.000	75.000	88.649
GEO7904_GEO29431	GEO15852	88.372	86.047	87.209	87.209
GEO7904_GEO15852	GEO29431	100.000	83.333	91.667	96.970
GEO7904_GEO29431	GEO42568	100.000	76.471	88.235	96.694
GEO42568_GEO10780	GEO7904	95.238	47.368	71.303	79.032
GEO9348_GEO23878	GEO10715	78.947	60.000	69.474	70.000
GEO10715_GEO23878	GEO20916	100.000	100.000	100.000	100.000
GEO24514_GEO9348	GEO23878	100.000	100.000	100.000	98.305
GEO9348_GEO20916	GEO24514	100.000	93.333	96.667	97.959
GEO10715_GEO20916	GEO9348	100.000	100.000	100.000	100.000
GEO19804_GEO18842	GEO10072	100.000	93.878	96.939	97.196
GEO19804_GEO10072	GEO18842	100.000	100.000	100.000	100.000
GEO10072_GEO7670	GEO19188	96.703	93.846	95.275	95.513
GEO7670_GEO18842	GEO19804	96.667	93.333	95.000	95.000
GEO19804_GEO18842	GEO7670	94.872	96.296	95.584	95.455
GEO6956_GEO7904	GEO4412	98.305	50.000	74.153	83.529
GEO7904_GEO9348	GEO6956	98.551	55.000	76.775	88.764
GEO7904_GEO4412	GEO7670	97.436	85.185	91.311	92.424
GEO7670_GEO4412	GEO7904	97.674	36.842	67.258	79.032
GEO6956_GEO7904	GEO9348	100.000	83.333	91.667	97.561
GEO46602_GEO32448	GEO17951	83.824	91.304	87.564	87.591
GEO82188_GEO17951	GEO32448	100.000	90.000	95.000	93.750
GEO6956_GEO32448	GEO46602	100.000	92.857	96.429	98.000
GEO46602_GEO32448	GEO6956	100.000	60.000	80.000	91.011
GEO6956_GEO46602	GEO82188	92.308	91.549	91.928	91.912

Table 79: Accuracy of MS-TRL with 3 best sources

Source	Target	SN	SP	BACC	AccAb
GEO1993_GEO4290_GEO4412	GEO16011	98.742	12.500	55.621	90.857
GEO4271_GEO16011_GEO4290	GEO1993	94.872	68.421	81.646	86.207
GEO1993_GEO16011_GEO4290	GEO4271	98.684	25.000	61.842	81.000
GEO1993_GEO4271_GEO16011	GEO4290	96.296	31.579	63.938	84.000
GEO1993_GEO4271_GEO4290	GEO4412	96.610	61.538	79.074	85.882
GEO7904_GEO42568_GEO29431	GEO10780	50.000	100.000	75.000	88.649
GEO7904_GEO10780_GEO29431	GEO15852	86.047	86.047	86.047	86.047
GEO7904_GEO15852_GEO42568	GEO29431	100.000	83.333	91.667	96.970
GEO7904_GEO15852_GEO29431	GEO42568	100.000	70.588	85.294	95.868
GEO15852_GEO42568_GEO10780	GEO7904	95.238	42.105	68.672	77.419
GEO9348_GEO20916_GEO23878	GEO10715	84.211	60.000	72.105	73.333
GEO9348_GEO10715_GEO23878	GEO20916	100.000	100.000	100.000	100.000
GEO24514_GEO10715_GEO20916	GEO23878	97.143	100.000	98.571	98.305
GEO9348_GEO10715_GEO20916	GEO24514	100.000	93.333	96.667	97.959
GEO24514_GEO10715_GEO20916	GEO9348	100.000	100.000	100.000	100.000
GEO19804_GEO18842_GEO19188	GEO10072	100.000	91.837	95.918	96.262
GEO19804_GEO10072_GEO7670	GEO18842	100.000	100.000	100.000	100.000
GEO10072_GEO7670_GEO18842	GEO19188	96.703	93.846	95.275	95.513
GEO10072_GEO18842_GEO19188	GEO19804	95.000	95.000	95.000	95.000
GEO19804_GEO10072_GEO18842	GEO7670	94.872	96.296	95.584	95.455
GEO6956_GEO7904_GEO7670	GEO4412	98.305	50.000	74.153	83.529
GEO7904_GEO9348_GEO7670	GEO6956	97.101	55.000	76.051	87.640
GEO7904_GEO9348_GEO4412	GEO7670	97.436	85.185	91.311	92.424
GEO6956_GEO7670_GEO4412	GEO7904	97.674	31.579	64.627	77.419
GEO6956_GEO7904_GEO4412	GEO9348	100.000	75.000	87.500	96.341
GEO6956_GEO46602_GEO32448	GEO17951	83.824	91.304	87.564	87.591
GEO46602_GEO82188_GEO17951	GEO32448	97.436	92.500	94.968	93.750
GEO6956_GEO82188_GEO32448	GEO46602	100.000	85.714	92.857	96.000
GEO82188_GEO17951_GEO32448	GEO6956	95.652	50.000	72.826	85.393
GEO6956_GEO46602_GEO32448	GEO82188	89.231	90.141	89.686	89.706

Table 80: Accuracy of MS-TRL with all sources

Source	Target	SN	SP	BACC	AccAb
all_sources	GEO16011	96.855	12.500	54.678	89.143
all_sources	GEO1993	92.308	68.421	80.364	84.483
all_sources	GEO4271	98.684	20.833	59.759	80.000
all_sources	GEO4290	95.062	31.579	63.320	83.000
all_sources	GEO4412	98.305	53.846	76.076	84.706
all_sources	GEO10780	47.619	100.000	73.810	88.108
all_sources	GEO15852	88.372	83.721	86.047	86.047
all_sources	GEO29431	98.148	83.333	90.741	95.455
all_sources	GEO42568	99.038	70.588	84.813	95.041
all_sources	GEO7904	93.023	42.105	67.564	77.419
all_sources	GEO10715	73.684	70.000	71.842	70.000
all_sources	GEO20916	97.222	97.059	97.141	97.143
all_sources	GEO23878	97.143	100.000	98.571	98.305
all_sources	GEO24514	100.000	93.333	96.667	97.959
all_sources	GEO9348	100.000	100.000	100.000	100.000
all_sources	GEO10072	100.000	91.837	95.918	95.327
all_sources	GEO18842	100.000	100.000	100.000	100.000
all_sources	GEO19188	96.703	92.308	94.505	94.872
all_sources	GEO19804	95.000	95.000	95.000	95.000
all_sources	GEO7670	94.872	92.593	93.732	93.939
all_sources	GEO4412	96.610	50.000	73.305	82.353
all_sources	GEO6956	95.652	55.000	75.326	86.517
all_sources	GEO7670	94.872	81.481	88.177	89.394
all_sources	GEO7904	97.674	31.579	64.627	77.419
all_sources	GEO9348	100.000	66.667	83.333	95.122
all_sources	GEO17951	86.765	85.507	86.136	86.131
all_sources	GEO32448	95.000	92.500	93.750	93.750
all_sources	GEO46602	97.222	85.714	91.468	94.000
all_sources	GEO6956	95.652	45.000	70.326	84.270
all_sources	GEO82188	89.231	90.141	89.686	89.706

E.2 ROBUST RULE PATTERNS VIA MS-TRL

E.2.1 BRAIN CANCER

```
=== TARGET ===
GEO1993.txt

=== SOURCES ===
S1: GEO4271.txt
S2: GEO16011.txt
S3: GEO4290.txt
S4: GEO4412.txt

=== Robust rules ===
S1,S4 12. IF(COL3A1 = -inf..7.259)) THEN (@Class = CONTROL)
   CF=0.904, PV=4.78375E-11, TP=27, FP=5, Pos=67, Neg=171
S1,S3 7. IF(PRKCZ = -inf..6.716)) THEN (@Class = CASE)
   CF=0.924, PV=5.30643E-5, TP=50, FP=1, Pos=193, Neg=60
S1,S4 14. IF(ABLIM1 = 9.413..inf)) THEN (@Class = CONTROL)
   CF=0.871, PV=2.01865E-11, TP=35, FP=11, Pos=67, Neg=170
S3,S4 4. IF(PSMA3 = 8.022..inf)) THEN (@Class = CASE)
   CF=0.977, PV=1.16612E-5, TP=42, FP=0, Pos=175, Neg=62
S1,S4 10. IF(SMC4 = -inf..6.228)) THEN (@Class = CONTROL)
   CF=0.905, PV=2.83551E-12, TP=31, FP=6, Pos=67, Neg=170
S1,S2,S3 4. IF(PRKCZ = -inf..6.802)) THEN (@Class = CASE)
   CF=0.945, PV=5.70458E-7, TP=95, FP=1, Pos=351, Neg=76
S1,S2 4. IF(MBD2 = 7.053..inf)) THEN (@Class = CASE)
   CF=0.976, PV=2.15297E-3, TP=39, FP=0, Pos=270, Neg=57
```

```
=== TARGET ===
GEO4271.txt

=== SOURCES ===
S1: GEO1993.txt
S2: GEO16011.txt
S3: GEO4290.txt
S4: GEO4412.txt

=== Robust rules ===
S3,S4 9. IF(NUCB1 = 10.347..inf)) THEN (@Class = CASE)
   CF=0.94, PV=1.27185E-6, TP=63, FP=1, Pos=208, Neg=67
S1,S4 10. IF(MGP = -inf..7.792)) THEN (@Class = CONTROL)
   CF=0.935, PV=5.35139E-12, TP=25, FP=2, Pos=67, Neg=166
S3,S4 14. IF(PPP2R5A = 10.062..inf)) THEN (@Class = CONTROL)
   CF=0.894, PV=5.16809E-13, TP=32, FP=9, Pos=67, Neg=208
S2,S3 13. IF(IGFBP4 = 9.201..inf)) THEN (@Class = CASE)
   CF=0.891, PV=3.48447E-5, TP=96, FP=2, Pos=308, Neg=57
S1,S3 3. IF(STXBP1 = -inf..9.626)) THEN (@Class = CASE)
   CF=0.937, PV=2.42759E-6, TP=61, FP=1, Pos=188, Neg=60
S2,S3 15. IF(MORF4L2 = 12.887..inf)) THEN (@Class = CASE)
   CF=0.88, PV=9.76961E-3, TP=46, FP=1, Pos=308, Neg=57
S1,S2,S4 16. IF(HSPG2 = 6.593..inf)) THEN (@Class = CASE)
   CF=0.838, PV=5.41672E-7, TP=126, FP=6, Pos=325, Neg=83
S1,S2 7. IF(NASP = 10.219..inf)) THEN (@Class = CASE)
   CF=0.979, PV=6.31374E-4, TP=46, FP=0, Pos=267, Neg=56
S2,S3 15. IF(PPP3CB = -inf..9.884)) THEN (@Class = CASE)
   CF=0.905, PV=9.39122E-9, TP=167, FP=3, Pos=309, Neg=56
```

```
=== TARGET ===
GEO4290.txt

=== SOURCES ===
S1: GEO1993.txt
S2: GEO4271.txt
S3: GEO16011.txt
S4: GEO4412.txt

=== Robust rules ===
S1,S4 16. IF(IFNGR2 = -inf..11.235)) THEN (@Class = CONTROL)
   CF=0.863, PV=2.9495E-10, TP=31, FP=11, Pos=63, Neg=170
S1,S3 1. IF(THBS1 = 6.373..inf)
             (RAE1 = 10.352..inf)) THEN (@Class = CASE)
   CF=0.991, PV=3.29205E-8, TP=108, FP=0, Pos=271, Neg=52
S2,S3 2. IF(SLC9A3R1 = -inf..9.608)
             (PPIF = -inf..10.553)) THEN (@Class = CASE)
   CF=0.989, PV=1.01364E-6, TP=91, FP=0, Pos=308, Neg=57
S2,S3,S4 12. IF(PPIB = 12.877..inf)) THEN (@Class = CASE)
   CF=0.919, PV=1.44682E-7, TP=110, FP=2, Pos=367, Neg=83
S1,S2 14. IF(ALDH2 = 13.026..inf)) THEN (@Class = CONTROL)
   CF=0.903, PV=1.22319E-11, TP=26, FP=6, Pos=60, Neg=188
S2,S4 5. IF(NET1 = -inf..9.021)) THEN (@Class = CASE)
   CF=0.944, PV=3.48346E-7, TP=68, FP=1, Pos=208, Neg=67
S1,S4 3. IF(ITPR3 = 7.331..inf)) THEN (@Class = CASE)
   CF=0.985, PV=1.61111E-8, TP=63, FP=0, Pos=171, Neg=62
S1,S3 10. IF(PFN1 = 12.861..inf)) THEN (@Class = CASE)
   CF=0.942, PV=1.86968E-6, TP=99, FP=1, Pos=271, Neg=52
S1,S4 14. IF(GLUD1 = 11.745..inf)
             (ACTA2 = -inf..10.827)) THEN (@Class = CONTROL)
```

CF=0.856, PV=6.02923E-11, TP=35, FP=14, Pos=62, Neg=171

S1,S2 15. IF(MYL9 = 8.413..inf)) THEN (@Class = CASE)

CF=0.851, PV=8.95858E-7, TP=94, FP=5, Pos=188, Neg=60

=== TARGET ===

GEO4412.txt

=== SOURCES ===

S1: GEO1993.txt

S2: GEO4271.txt

S3: GEO16011.txt

S4: GEO4290.txt

=== Robust rules ===

S1,S4 4. IF(ITPR3 = 9.381..inf)) THEN (@Class = CASE)

CF=0.98, PV=2.11915E-6, TP=48, FP=0, Pos=173, Neg=61

S1,S2 6. IF(MGP = -inf..9.736)) THEN (@Class = CONTROL)

CF=0.92, PV=6.49589E-11, TP=24, FP=3, Pos=66, Neg=168

S2,S4 8. IF(ACO2 = -inf..12.967)) THEN (@Class = CASE)

CF=0.944, PV=3.97578E-7, TP=69, FP=1, Pos=210, Neg=66

S2,S3 14. IF(RPS24 = -inf..17.042)) THEN (@Class = CASE)

CF=0.916, PV=3.18315E-7, TP=109, FP=2, Pos=288, Neg=63

S2,S4 5. IF(RAD21 = -inf..12.466)) THEN (@Class = CASE)

CF=0.943, PV=5.12497E-7, TP=68, FP=1, Pos=210, Neg=66

S3,S4 3. IF(GSS = 11.868..inf)) THEN (@Class = CASE)

CF=0.978, PV=1.22777E-3, TP=44, FP=0, Pos=293, Neg=59

S2,S4 15. IF(SNX19 = -inf..11.023)) THEN (@Class = CASE)

CF=0.889, PV=4.34601E-5, TP=57, FP=2, Pos=210, Neg=67

S1,S2,S4 18. IF(PPP2R5A = 12.456..inf)) THEN (@Class = CONTROL)

```
   CF=0.9, PV=3.21965E-15, TP=39, FP=10, Pos=86, Neg=249
S3,S4 7. IF(CASP3 = 10.998..inf)) THEN (@Class = CASE)
   CF=0.981, PV=4.28914E-4, TP=50, FP=0, Pos=293, Neg=59
S2,S4 8. IF(SNX19 = -inf..11.023)) THEN (@Class = CASE)
   CF=0.93, PV=1.445E-5, TP=54, FP=1, Pos=210, Neg=67
S2,S3 2. IF(ZMYM2 = -inf..11.957)) THEN (@Class = CASE)
   CF=0.986, PV=3.12389E-6, TP=71, FP=0, Pos=288, Neg=64
S1,S3 9. IF(UPP1 = 9.905..inf)) THEN (@Class = CASE)
   CF=0.935, PV=8.80857E-6, TP=74, FP=1, Pos=251, Neg=59
S2 15. IF(LAMC1 = 12.362..inf)) THEN (@Class = CASE)
   CF=0.89, PV=5.23229E-7, TP=72, FP=3, Pos=129, Neg=48
S1,S2 16. IF(ABLIM1 = 14.070..inf)) THEN (@Class = CONTROL)
   CF=0.858, PV=1.96977E-10, TP=34, FP=12, Pos=67, Neg=168
S1,S2 16. IF(SMC4 = -inf..11.376)) THEN (@Class = CONTROL)
   CF=0.843, PV=1.17172E-10, TP=38, FP=16, Pos=66, Neg=169


=== TARGET ===
GEO16011.txt

=== SOURCES ===
S1: GEO1993.txt
S2: GEO4271.txt
S3: GEO4290.txt
S4: GEO4412.txt

=== Robust Rules ===
S1,S3,S4 3. IF(PLOD1 = 8.723..inf)
         (ARPC1A = 10.395..inf)) THEN (@Class = CASE)
   CF=0.971, PV=2.5091E-14, TP=168, FP=1, Pos=322, Neg=78
```

```
S2,S3 16. IF(EPAS1 = 9.104..inf)
         (GLUD1 = -inf..10.090)) THEN (@Class = CASE)
   CF=0.876, PV=2.78827E-9, TP=192, FP=5, Pos=300, Neg=57
S1,S2 8. IF(PLOD1 = 8.723..inf) (IMMT = 9.917..inf)
                (MYL9 = 5.967..inf)) THEN (@Class = CASE)
   CF=0.906, PV=9.15999E-9, TP=139, FP=3, Pos=258, Neg=57
S1,S2,S3,S4 10. IF(CALU = 8.728..inf)) THEN (@Class = CASE)
   CF=0.88, PV=2.84046E-11, TP=178, FP=6, Pos=398, Neg=102
S2,S4 12. IF(LAMC1 = -inf..8.484)
          (IMMT = -inf..9.917)) THEN (@Class = CONTROL)
   CF=0.838, PV=2.68817E-8, TP=22, FP=15, Pos=64, Neg=278
S1,S2 1. IF(MYL9 = 5.970..inf)
          (C5orf13 = 8.444..inf)) THEN (@Class = CASE)
   CF=0.994, PV=1.84852E-13, TP=154, FP=0, Pos=258, Neg=58
S2,S3,S4 4. IF(LDHA = 12.680..inf)) THEN (@Class = CASE)
   CF=0.933, PV=1.00142E-13, TP=192, FP=3, Pos=359, Neg=84
S1,S2 5. IF(PTPRF = 7.792..inf)
           (MYL9 = 5.709..inf)) THEN (@Class = CASE)
   CF=0.935, PV=5.32095E-10, TP=141, FP=2, Pos=258, Neg=58
S1,S4 9. IF(PLOD1 = 8.723..inf)
           (MYL9 = 5.709..inf)) THEN (@Class = CASE)
   CF=0.903, PV=8.79279E-11, TP=158, FP=4, Pos=241, Neg=60
S2,S4 3. IF(LAMC1 = 9.107..inf)) THEN (@Class = CASE)
   CF=0.935, PV=4.75264E-10, TP=139, FP=2, Pos=279, Neg=64
```

E.2.2 BREAST CANCER

```
=== TARGET ===
```

GEO7904.txt

=== SOURCES ===

S1: GEO15852.txt

S2: GEO42568.txt

S3: GEO10780.txt

S4: GEO29431.txt

=== Robust Rules ===

S2,S4 1. IF(ACSL5 = -inf..6.050)) THEN (@Class = CASE)

CF=0.984, PV=9.63608E-6, TP=62, FP=0, Pos=196, Neg=46

S1,S2,S4 4. IF(ADH1B = 9.624..inf)) THEN (@Class = CONTROL)

CF=0.96, PV=0E0, TP=50, FP=3, Pos=90, Neg=239

S1,S2 8. IF(ADIPOQ = 7.488..inf)) THEN (@Class = CONTROL)

CF=0.912, PV=0E0, TP=48, FP=9, Pos=77, Neg=185

S2,S4 9. IF(ACADS = 7.037..inf)) THEN (@Class = CONTROL)

CF=0.909, PV=4.21885E-15, TP=30, FP=9, Pos=46, Neg=196

S1,S2,S4 9. IF(ANGPT1 = -inf..5.018)) THEN (@Class = CASE)

CF=0.905, PV=3.57936E-13, TP=137, FP=5, Pos=240, Neg=89

S2,S4 7. IF(ACAA2 = -inf..8.449)) THEN (@Class = CASE)

CF=0.907, PV=2.30744E-6, TP=92, FP=2, Pos=197, Neg=46

S1,S2,S4 2. IF(ABCA8 = 8.548..inf)) THEN (@Class = CONTROL)

CF=0.957, PV=0E0, TP=38, FP=2, Pos=89, Neg=240

=== TARGET ===

GEO10780.txt

=== SOURCES ===

S1: GEO7904.txt

S2: GEO15852.txt

S3: GEO42568.txt

S4: GEO29431.txt

== Robust Rules ===

S2,S3 5. IF(ALDH18A1 = -inf..8.064)) THEN (@Class = CONTROL)

CF=0.88, PV=0E0, TP=104, FP=13, Pos=189, Neg=185

S1,S2,S3,S4 6. IF(ADH1B = 11.517..inf)) THEN (@Class = CONTROL)

CF=0.807, PV=2.36592E-8, TP=52, FP=15, Pos=220, Neg=282

S2,S3 4. IF(CSK = -inf..8.147)) THEN (@Class = CONTROL)

CF=0.897, PV=0E0, TP=114, FP=12, Pos=189, Neg=185

=== TARGET ===

GEO15852.txt

=== SOURCES ===

S1: GEO7904.txt

S2: GEO42568.txt

S3: GEO10780.txt

S4: GEO29431.txt

=== Robust Rules ===

S1,S2 5. IF(ADH1C = 6.888..inf)) THEN (@Class = CONTROL)

CF=0.95, PV=0E0, TP=41, FP=3, Pos=74, Neg=186

S2,S4 8. IF(ACACB = -inf..8.532)) THEN (@Class = CASE)

CF=0.924, PV=0E0, TP=186, FP=5, Pos=197, Neg=68

S2,S4 7. IF(ACACB = 8.532..inf)) THEN (@Class = CONTROL)

CF=0.929, PV=0E0, TP=62, FP=11, Pos=68, Neg=197

```
=== TARGET ===
GEO29431.txt

=== SOURCES ===
S1: GEO7904.txt
S2: GEO15852.txt
S3: GEO42568.txt
S4: GEO10780.txt

=== Robust Rules ===
S2,S3 5. IF(COL11A1 = 7.688..inf)) THEN (@Class = CASE)
   CF=0.955, PV=9.99201E-16, TP=139, FP=2, Pos=196, Neg=70
S2,S3 5. IF(ECT2 = 7.814..inf)) THEN (@Class = CASE)
   CF=0.973, PV=2.22045E-16, TP=134, FP=1, Pos=196, Neg=71
S1,S2 7. IF(ADH1B = 11.084..inf)) THEN (@Class = CONTROL)
   CF=0.959, PV=4.55191E-15, TP=35, FP=1, Pos=73, Neg=135
S2,S3 4. IF(ASPA = 8.321..inf)) THEN (@Class = CONTROL)
   CF=0.965, PV=0E0, TP=47, FP=2, Pos=71, Neg=196


=== TARGET ===
GEO42568.txt

=== SOURCES ===
S1: GEO7904.txt
S2: GEO15852.txt
S3: GEO10780.txt
S4: GEO29431.txt
```

```
=== Robust Rules ===
S1,S2 5. IF(COMP = 6.190..inf)) THEN (@Class = CASE)
   CF=0.953, PV=4.32987E-15, TP=114, FP=2, Pos=179, Neg=77
S2,S4 4. IF(G0S2 = 12.039..inf)) THEN (@Class = CONTROL)
   CF=0.968, PV=0E0, TP=40, FP=1, Pos=70, Neg=191
S1,S2 6. IF(FN1 = 11.440..inf)) THEN (@Class = CASE)
   CF=0.938, PV=5.55112E-15, TP=121, FP=3, Pos=180, Neg=77
```

E.2.3 COLON CANCER

```
 === TARGET ===
GEO9348.txt

=== SOURCES ===
S1: GEO24514.txt
S2: GEO10715.txt
S3: GEO20916.txt
S4: GEO23878.txt

=== Robust rules ===
S1,S4 3. IF(HILPDA = -inf..7.338)) THEN (@Class = CONTROL)
   CF=0.98, PV=0E0, TP=48, FP=0, Pos=50, Neg=132
S3,S4 1. IF(CDH3 = 5.874..inf)) THEN (@Class = CASE)
   CF=0.993, PV=0E0, TP=133, FP=0, Pos=134, Neg=69
S3 2. IF(CDH3 = -inf..5.874)) THEN (@Class = CONTROL)
   CF=0.979, PV=0E0, TP=45, FP=0, Pos=45, Neg=99
```

```
=== TARGET ===
GEO10715.txt

=== SOURCE ===
S1: GEO24514.txt
S2: GEO9348.txt
S3: GEO20916.txt
S4: GEO23878.txt

=== Robust rules ===
S1,S3 1. IF(CNNM2 = 6.231..inf)) THEN (@Class = CONTROL)
   CF=0.966, PV=0E0, TP=46, FP=1, Pos=59, Neg=87



=== TARGET ===
GEO20916.txt

=== SOURCES ===
S1: GEO24514.txt
S2: GEO9348.txt
S3: GEO10715.txt
S4: GEO23878.txt

=== Robust Rules ===
S4 3. IF(HILPDA = 8.429..inf)) THEN (@Class = CASE)
   CF=0.963, PV=1.96287E-13, TP=56, FP=1, Pos=67, Neg=55
S1,S4 1. IF(HILPDA = -inf..7.901)) THEN (@Class = CONTROL)
   CF=0.983, PV=0E0, TP=58, FP=0, Pos=70, Neg=101

=== TARGET ===
```

GEO23878.txt

=== SOURCES ===

S1: GEO24514.txt

S2: GEO9348.txt

S3: GEO10715.txt

S4: GEO20916.txt

=== Robust rules ===

S2,S4 2. IF(CA7 = 6.663..inf)) THEN (@Class = CONTROL)

   CF=0.984, PV=0E0, TP=61, FP=0, Pos=67, Neg=138

S2 4. IF(CA7 = -inf..6.663)) THEN (@Class = CASE)

   CF=0.935, PV=4.48012E-10, TP=102, FP=2, Pos=102

S1 4. IF(ABCA8 = 7.290..inf)) THEN (@Class = CONTROL)

   CF=0.971, PV=1.26565E-14, TP=32, FP=0, Pos=37, Neg=66

Nr 7. IF(ABCA8 = -inf..7.290)) THEN (@Class = CASE)

   CF=0.971, PV=1.7356E-7, TP=32, FP=0, Pos=32, Neg=22

=== TARGET ===

GEO24514.txt

=== SOURCES ===

S1: GEO9348.txt

S2: GEO10715.txt

S3: GEO20916.txt

S4: GEO23878.txt

=== Robust rules ===

S3,S4 1. IF(CDH3 = 7.449..inf)) THEN (@Class = CASE)

   CF=0.977, PV=0E0, TP=101, FP=1, Pos=102, Neg=71

S3,S4 2. IF(CDH3 = -inf..7.449)) THEN (@Class = CONTROL)

CF=0.977, PV=0E0, TP=70, FP=1, Pos=71, Neg=102

E.2.4 LUNG CANCER

=== TARGET ===

GEO7670.txt

=== SOURCES ===

S1: GEO19804.txt

S2: GEO10072.txt

S3: GEO18842.txt

S4: GEO19188.txt

=== Robust Rules ===

S1 8. IF(AHNAK = 12.145..inf)) THEN (@Class = CONTROL)

CF=0.969, PV=0E0, TP=58, FP=1, Pos=84, Neg=95

Nr 9. IF(AHNAK = -inf..12.175)) THEN (@Class = CASE)

CF=0.972, PV=6.48352E-8, TP=34, FP=0, Pos=35

S1,S4 5. IF(EMP2 = 11.747..inf)) THEN (@Class = CONTROL)

CF=0.972, PV=0E0, TP=116, FP=3, Pos=149, Neg=186

=== TARGET ===

GEO10072.txt

=== SOURCES ===

S1: GEO19804.txt

S2: GEO7670.txt

```
S3: GEO18842.txt
S4: GEO19188.txt

=== Robust Rules ===
S2,S4 6. IF(CLIC5 = -inf..9.439)) THEN (@Class = CASE)
   CF=0.978, PV=0E0, TP=163, FP=2, Pos=182, Neg=136
S1,S4 7. IF(ARHGEF15 = 8.265..inf)) THEN (@Class = CONTROL)
   CF=0.978, PV=0E0, TP=117, FP=2, Pos=169, Neg=203
S1,S2,S4 6. IF(FAM107A = -inf..8.803)) THEN (@Class = CASE)
   CF=0.973, PV=0E0, TP=209, FP=4, Pos=242, Neg=197
S1,S2 6. IF(FAM189A2 = 8.816..inf)) THEN (@Class = CONTROL)
   CF=0.973, PV=0E0, TP=99, FP=2, Pos=131, Neg=15
S1,S2,S4 6. IF(AGER = 8.923..inf)) THEN (@Class = CONTROL)
   CF=0.967, PV=0E0, TP=171, FP=6, Pos=196, Neg=243
S1,S4 5. IF(EFNA4 = 5.989..inf)) THEN (@Class = CASE)
   CF=0.971, PV=0E0, TP=152, FP=3, Pos=204, Neg=169


=== TARGET ===
GEO18842.txt

=== SOURCES ===
S1: GEO19804.txt
S2: GEO10072.txt
S3: GEO7670.txt
S4: GEO19188.txt

=== Robust Rules ===
S4 1. IF(AQP1 = -inf..10.949)) THEN (@Class = CASE)
   CF=0.992, PV=0E0, TP=119, FP=0, Pos=132, Neg=106
```

```
Nr 6. IF(AQP4 = 10.455..inf)) THEN (@Class = CONTROL)
   CF=0.977, PV=1.45239E-12, TP=41, FP=0, Pos=41, Neg=41
S1,S2 1. IF(FAM107A = -inf..8.559)) THEN (@Class = CASE)
   CF=0.992, PV=0E0, TP=116, FP=0, Pos=160, Neg=149
S2,S3 1. IF(EDNRB = -inf..7.750)) THEN (@Class = CASE)
   CF=0.993, PV=0E0, TP=134, FP=0, Pos=139, Neg=116
S2 7. IF(EDNRB = 7.750..inf)) THEN (@Class = CONTROL)
   CF=0.97, PV=0E0, TP=89, FP=2, Pos=89, Neg=100


=== TARGET ===
GEO19188.txt

=== SOURCES ===
S1: GEO19804.txt
S2: GEO10072.txt
S3: GEO7670.txt
S4: GEO18842.txt

=== Robust rules ===
S2,S3 1. IF(EDNRB = -inf..8.107)) THEN (@Class = CASE)
   CF=0.994, PV=0E0, TP=170, FP=0, Pos=179, Neg=134


=== TARGET ===
GEO19804.txt

=== SOURCES ===
S1: GEO10072.txt
S2: GEO7670.txt
```

```
S3: GEO18842.txt
S4: GEO19188.txt

=== Robust Rules ===
S2 2. IF(FAM189A2 = 8.240..inf)) THEN (@Class = CONTROL)
   CF=0.972, PV=0E0, TP=64, FP=1, Pos=81, Neg=93
S4 5. IF(FAM107A = -inf..8.850)) THEN (@Class = CASE)
   CF=0.964, PV=0E0, TP=124, FP=3, Pos=145, Neg=119
Nr 17. IF(FAM189A1 = -inf..4.560)) THEN (@Class = CASE)
   CF=0.973, PV=8.67973E-11, TP=35, FP=0, Pos=54, Neg=54
S2 6. IF(AGER = -inf..9.445)) THEN (@Class = CASE)
   CF=0.976, PV=0E0, TP=88, FP=1, Pos=93, Neg=81
S1 7. IF(AGER = 9.445..inf)) THEN (@Class = CONTROL)
   CF=0.972, PV=0E0, TP=96, FP=2, Pos=103, Neg=112
S2,S4 6. IF(FABP4 = -inf..8.791)) THEN (@Class = CASE)
   CF=0.962, PV=0E0, TP=152, FP=4, Pos=184, Neg=146
S4 7. IF(ALDH3B2 = 4.816..inf)) THEN (@Class = CASE)
   CF=0.968, PV=3.33067E-16, TP=67, FP=1, Pos=145
S1 11. IF(ALDH18A1 = -inf..8.158)) THEN (@Class = CONTROL)
   CF=0.959, PV=0E0, TP=86, FP=3, Pos=103, Neg=112
```

E.2.5 MIX CANCER

```
 === TARGET ===
GEO4412.txt

=== SOURCES ===
S1: GEO6956.txt
S2: GEO7904.txt
```

S3: GEO9348.txt

S4: GEO7670.txt

=== Robust Rules ===

S1,S2 7. IF(CASP8 = 9.728..inf)) THEN (@Class = CASE)

CF=0.92, PV=7.3607E-5, TP=41, FP=1, Pos=165, Neg=62

S2,S4 9. IF(ABLIM1 = 14.364..inf)) THEN (@Class = CONTROL)

CF=0.922, PV=2.90551E-9, TP=23, FP=2, Pos=70, Neg=135

S3,S4 2. IF(CALU = 13.113..inf)) THEN (@Class = CASE)

CF=0.93, PV=7.83928E-13, TP=115, FP=3, Pos=162, Neg=63

Nr 24. IF(COL5A2 = -inf..10.045)) THEN (@Class = CONTROL)

CF=0.923, PV=3.23118E-6, TP=11, FP=0, Pos=24, Neg=53

S2,S4 4. IF(COL5A1 = 9.352..inf)) THEN (@Class = CASE)

CF=0.914, PV=9.57892E-10, TP=71, FP=3, Pos=135, Neg=70

=== TARGET ===

GEO6956.txt

=== SOURCES ===

S1: GEO7904.txt

S2: GEO9348.txt

S3: GEO7670.txt

S4: GEO4412.txt

=== Robust Rules ===

S1,S4 6. IF(BAG1 = -inf..8.945)) THEN (@Class = CASE)

CF=0.882, PV=2.10034E-6, TP=65, FP=3, Pos=164, Neg=63

S1,S2 1. IF(ACADS = 6.763..inf)) THEN (@Class = CONTROL)

CF=0.947, PV=3.33067E-16, TP=27, FP=2, Pos=49, Neg=175

S1,S4 4. IF(DHX9 = -inf..8.370)) THEN (@Class = CASE)

CF=0.899, PV=8.43334E-6, TP=54, FP=2, Pos=164, Neg=63

=== TARGET ===

GEO7670.txt

=== SOURCES ===

S1: GEO6956.txt

S2: GEO7904.txt

S3: GEO9348.txt

S4: GEO4412.txt

=== Robust Rules ===

S1,S4 1. IF(CGRRF1 = -inf..6.787)) THEN (@Class = CASE)

CF=0.978, PV=8.43651E-7, TP=43, FP=0, Pos=163, Neg=70

S3 4. IF(GARS = 10.044..inf)) THEN (@Class = CASE)

CF=0.963, PV=1.28723E-11, TP=101, FP=1, Pos=105, Neg=36

S3 5. IF(GARS = -inf..10.044)) THEN (@Class = CONTROL)

CF=0.938, PV=0E0, TP=35, FP=4, Pos=36, Neg=105

=== TARGET ===

GEO7904.txt

=== SOURCES ===

S1: GEO6956.txt

S2: GEO9348.txt

S3: GEO7670.txt

S4: GEO4412.txt

```
=== Robust Rules ===
S2,S3,S4 6. IF(ABCG2 = -inf..5.029)) THEN (@Class = CASE)
   CF=0.901, PV=2.59581E-12, TP=122, FP=5, Pos=206, Neg=82
S3,S4 7. IF(AKT3 = -inf..7.035)) THEN (@Class = CASE)
   CF=0.886, PV=4.69261E-5, TP=37, FP=2, Pos=136, Neg=70
S1,S2 4. IF(ACADS = 7.037..inf)) THEN (@Class = CONTROL)
   CF=0.947, PV=2.22045E-16, TP=27, FP=2, Pos=49, Neg=177
S1 2. IF(ACADS = -inf..7.037) (ADAMTS2 = 7.133..inf)) THEN (@Class = CASE)
   CF=0.971, PV=2.00021E-4, TP=33, FP=0, Pos=107, Neg=37
S1,S3 4. IF(ANGPT1 = -inf..5.163)) THEN (@Class = CASE)
   CF=0.94, PV=5.03122E-7, TP=51, FP=1, Pos=147, Neg=64
S1,S2,S3 5. IF(ACAA2 = -inf..8.449)) THEN (@Class = CASE)
   CF=0.941, PV=6.25469E-7, TP=61, FP=1, Pos=217, Neg=76
S3,S4 6. IF(ABLIM1 = 10.319..inf)) THEN (@Class = CONTROL)
   CF=0.93, PV=1.167E-10, TP=26, FP=2, Pos=70, Neg=137
```

E.2.6 PROSTATE CANCER

```
 === TARGET ===
GEO6956.txt

=== SOURCES ===
S1: GEO46602.txt
S2: GEO82188.txt
S3: GEO17951.txt
S4: GEO32448.txt

=== Robust rules ===
```

S2,S3 3. IF(E2F5 = -inf..5.024)) THEN (@Class = CONTROL)

CF=0.949, PV=1.12626E-10, TP=33, FP=1, Pos=158, Neg=195

S1,S3 6. IF(DPT = -inf..6.315)) THEN (@Class = CASE)

CF=0.937, PV=8.38427E-11, TP=63, FP=2, Pos=166, Neg=101

S1,S4 10. IF(CSTA = 9.448..inf)) THEN (@Class = CONTROL)

CF=0.924, PV=2.55351E-15, TP=43, FP=5, Pos=72, Neg=138

S2,S3 17. IF(BDH1 = -inf..6.859)) THEN (@Class = CONTROL)

CF=0.882, PV=1.92069E-14, TP=67, FP=10, Pos=158, Neg=195

S2,S3 19. IF(ANP32E = 9.329..inf)) THEN (@Class = CONTROL)

CF=0.86, PV=2.40164E-9, TP=45, FP=8, Pos=158, Neg=195

S2,S3 4. IF(FEZ2 = 10.575..inf)) THEN (@Class = CONTROL)

CF=0.958, PV=5.23803E-13, TP=40, FP=1, Pos=158, Neg=195

S2,S3 15. IF(GDF15 = -inf..9.522)) THEN (@Class = CONTROL)

CF=0.877, PV=0E0, TP=99, FP=16, Pos=158, Neg=195

S2,S3 16. IF(GRSF1 = 8.869..inf)) THEN (@Class = CASE)

CF=0.874, PV=2.44249E-15, TP=100, FP=11, Pos=195, Neg=158

S2,S3 7. IF(CAMKK2 = 10.412..inf)) THEN (@Class = CASE)

CF=0.934, PV=1.10245E-13, TP=66, FP=3, Pos=195, Neg=158

S2,S3 12. IF(APOL1 = -inf..8.203)) THEN (@Class = CASE)

CF=0.917, PV=1.38511E-12, TP=65, FP=4, Pos=195, Neg=158

S2,S3 1. IF(DAPK1 = 9.127..inf)) THEN (@Class = CASE)

CF=0.987, PV=0E0, TP=74, FP=0, Pos=195, Neg=158

S1,S2,S3,S4 12. IF(ANXA2 = -inf..12.361)) THEN (@Class = CASE)

CF=0.916, PV=0E0, TP=142, FP=9, Pos=231, Neg=172

S1,S4 15. IF(FLRT3 = 7.820..inf)) THEN (@Class = CONTROL)

CF=0.881, PV=6.15064E-13, TP=45, FP=10, Pos=72, Neg=138

S2,S3 4. IF(CDC42BPA = 9.773..inf)) THEN (@Class = CONTROL)

CF=0.937, PV=2.35821E-11, TP=38, FP=2, Pos=158, Neg=195

S2,S3,S4 12. IF(DNAJC15 = -inf..8.187)) THEN (@Class = CASE)

CF=0.902, PV=2.10942E-15, TP=85, FP=7, Pos=235, Neg=198

S1,S2 7. IF(DUS1L = -inf..9.044)) THEN (@Class = CONTROL)

CF=0.919, PV=2.08322E-12, TP=39, FP=4, Pos=103, Neg=163

S1,S3 10. IF(BEX1 = -inf..6.644)) THEN (@Class = CASE)

CF=0.945, PV=1.57863E-12, TP=72, FP=2, Pos=166, Neg=101

S1,S2,S3 6. IF(HPN = 8.074..inf)) THEN (@Class = CASE)

CF=0.924, PV=0E0, TP=174, FP=10, Pos=231, Neg=172

=== TARGET ===

GEO17951.txt

=== SOURCES ===

S1: GEO6956.txt

S2: GEO46602.txt

S3: GEO82188.txt

S4: GEO32448.txt

=== Robust rules ===

S1,S3 18. IF(ANXA2 = -inf..11.597)) THEN (@Class = CASE)

CF=0.942, PV=4.44089E-16, TP=78, FP=3, Pos=195, Neg=153

S3 10. IF(ANXA6 = 9.700..inf)) THEN (@Class = CONTROL)

CF=0.953, PV=1.33227E-15, TP=62, FP=2, Pos=133, Neg=126

S3,S4 17. IF(ABCC4 = 11.120..inf)) THEN (@Class = CASE)

CF=0.928, PV=0E0, TP=98, FP=7, Pos=166, Neg=173

S1,S3 21. IF(CLDN3 = 8.384..inf)) THEN (@Class = CASE)

CF=0.892, PV=1.86295E-13, TP=81, FP=7, Pos=195

S2,S4 21. IF(AMACR = -inf..8.466)) THEN (@Class = CONTROL)

CF=0.895, PV=0E0, TP=80, FP=10, Pos=116, Neg=137

S3,S4 6. IF(B3GAT1 = 7.493..inf)) THEN (@Class = CASE)

CF=0.971, PV=0E0, TP=65, FP=1, Pos=166, Neg=173

```
S2,S4 14. IF(COL4A6 = 8.302..inf)) THEN (@Class = CONTROL)
   CF=0.909, PV=4.44089E-16, TP=68, FP=7, Pos=116, Neg=137
S2 1. IF(COL4A6 = -inf..7.338)) THEN (@Class = CASE)
   CF=0.983, PV=1.90958E-14, TP=57, FP=0, Pos=97, Neg=76
S1,S3,S4 2. IF(ERG = 8.766..inf)) THEN (@Class = CASE)
   CF=0.973, PV=0E0, TP=79, FP=1, Pos=236, Neg=193
S1,S3 4. IF(C2orf72 = 8.250..inf)) THEN (@Class = CASE)
   CF=0.953, PV=2.22045E-15, TP=71, FP=2, Pos=196, Neg=153
S1,S2 8. IF(AOX1 = -inf..7.045)) THEN (@Class = CASE)
   CF=0.955, PV=7.77156E-16, TP=92, FP=2, Pos=166, Neg=97
S1,S3 16. IF(GOLM1 = 11.838..inf)) THEN (@Class = CASE)
   CF=0.909, PV=0E0, TP=97, FP=7, Pos=195, Neg=154

=== TARGET ===
GEO32448.txt

=== SOURCES ===
S1: GEO6956.txt
S2: GEO46602.txt
S3: GEO82188.txt
S4: GEO17951.txt

=== Robust rules ===
S1,S2,S3,S4 15. IF(ANXA2 = -inf..10.596)) THEN (@Class = CASE)
   CF=0.881, PV=0E0, TP=141, FP=14, Pos=274, Neg=210
S3,S4 1. IF(ABCC4 = 10.185..inf) (ANXA1 = -inf..8.243)) THEN (@Class = CASE)
   CF=0.973, PV=0E0, TP=69, FP=1, Pos=169, Neg=176
S2,S3,S4 4. IF(AKR1B1 = -inf..5.854)) THEN (@Class = CASE)
   CF=0.957, PV=8.52873E-12, TP=45, FP=1, Pos=205, Neg=190
S1,S2,S3,S4 5. IF(ACSM1 = 6.041..inf)) THEN (@Class = CASE)
```

CF=0.95, PV=1.94289E-14, TP=68, FP=2, Pos=274, Neg=210

S3,S4 7. IF(ABCC4 = 10.185..inf) (AKAP7 = -inf..7.068)) THEN (@Class = CASE)

CF=0.947, PV=0E0, TP=86, FP=4, Pos=169, Neg=176

S3,S4 17. IF(ADCY2 = -inf..6.747)) THEN (@Class = CASE)

CF=0.866, PV=2.868E-10, TP=55, FP=8, Pos=169, Neg=176

S1,S2,S4 5. IF(AHCY = 7.303..inf)) THEN (@Class = CASE)

CF=0.942, PV=6.1919E-8, TP=40, FP=1, Pos=209, Neg=139

S2,S4 11. IF(AMACR = -inf..5.492)) THEN (@Class = CONTROL)

CF=0.891, PV=1.76126E-8, TP=35, FP=4, Pos=119, Neg=140

S1,S4 14. IF(AMPD3 = 8.561..inf)) THEN (@Class = CASE)

CF=0.861, PV=4.5244E-5, TP=31, FP=3, Pos=173, Neg=125

S2,S3 6. IF(AKR1B1 = -inf..5.854)) THEN (@Class = CASE)

CF=0.948, PV=1.59185E-9, TP=38, FP=1, Pos=137, Neg=121

S1,S2,S4 7. IF(AHCY = 7.303..inf)) THEN (@Class = CASE)

CF=0.945, PV=2.38132E-8, TP=42, FP=1, Pos=209, Neg=139

S2,S3 6. IF(AGAP1 = 7.378..inf)) THEN (@Class = CASE)

CF=0.957, PV=7.09788E-12, TP=47, FP=1, Pos=137, Neg=121

S1,S2,S4 16. IF(AOX1 = -inf..6.525)) THEN (@Class = CASE)

CF=0.808, PV=3.91866E-8, TP=79, FP=12, Pos=209, Neg=139

S3,S4 9. IF(ACSF2 = -inf..5.928) (APOO = 6.659..inf)) THEN (@Class = CASE)

CF=0.923, PV=1.48881E-13, TP=57, FP=4, Pos=169, Neg=176

S2,S3 15. IF(ACSF2 = 6.073..inf)) THEN (@Class = CONTROL)

CF=0.828, PV=1.31091E-9, TP=59, FP=13, Pos=121, Neg=137

=== TARGET ===

GEO46602.txt

=== SOURCES ===

S1: GEO6956.txt

S2: GEO82188.txt

S3: GEO17951.txt

S4: GEO32448.txt

=== Robust rules ===

S2,S4 6. IF(CHRM3 = 6.647..inf)) THEN (@Class = CASE)

CF=0.962, PV=1.01918E-13, TP=53, FP=1, Pos=137, Neg=124

S2,S3 6. IF(ACTG2 = 11.680..inf)) THEN (@Class = CONTROL)

CF=0.956, PV=4.70046E-12, TP=41, FP=1, Pos=153, Neg=165

S2,S4 8. IF(CYP3A5 = 6.786..inf)) THEN (@Class = CONTROL)

CF=0.953, PV=2.36047E-11, TP=38, FP=1, Pos=124, Neg=137

S2,S3 5. IF(COBLL1 = 9.160..inf)) THEN (@Class = CASE)

CF=0.944, PV=1.16751E-12, TP=52, FP=2, Pos=165, Neg=153

S1,S3 6. IF(ARMCX1 = -inf..8.053)) THEN (@Class = CASE)

CF=0.96, PV=8.77842E-12, TP=64, FP=1, Pos=170, Neg=101

S1,S2,S3,S4 8. IF(B3GAT1 = 6.004..inf)) THEN (@Class = CASE)

CF=0.955, PV=2.22045E-16, TP=76, FP=2, Pos=275, Neg=212

S2,S4 9. IF(COL4A6 = 7.734..inf)) THEN (@Class = CONTROL)

CF=0.939, PV=4.77396E-15, TP=56, FP=3, Pos=123, Neg=138

S2,S4 1. IF(AKR1B1 = -inf..7.750)) THEN (@Class = CASE)

CF=0.98, PV=3.47611E-13, TP=47, FP=0, Pos=138, Neg=123

S1,S3 4. IF(AOX1 = -inf..6.249)) THEN (@Class = CASE)

CF=0.968, PV=4.32987E-15, TP=81, FP=1, Pos=170, Neg=101

S2,S3 4. IF(CBLC = -inf..5.543)) THEN (@Class = CONTROL)

CF=0.964, PV=1.1718E-8, TP=26, FP=0, Pos=152, Neg=166

=== TARGET ===

GEO82188.txt

```
=== SOURCES ===
S1: GEO6956.txt
S2: GEO46602.txt
S3: GEO17951.txt
S4: GEO32448.txt


=== Robust Rules ===
S1,S3,S4 8. IF(CSRP2 = -inf..8.918)) THEN (@Class = CASE)
   CF=0.938, PV=0E0, TP=108, FP=5, Pos=236, Neg=192
S1,S2 13. IF(ACACA = 10.223..inf)) THEN (@Class = CASE)
   CF=0.917, PV=8.84848E-13, TP=85, FP=4, Pos=164, Neg=97
S1,S3 1. IF(COX7A1 = -inf..8.432)) THEN (@Class = CASE)
   CF=0.986, PV=0E0, TP=68, FP=0, Pos=195, Neg=153
S3,S4 14. IF(EFEMP2 = 9.748..inf)) THEN (@Class = CONTROL)
   CF=0.918, PV=1.11022E-16, TP=80, FP=6, Pos=173, Neg=166
S3,S4 20. IF(BPHL = 7.094..inf)) THEN (@Class = CASE)
   CF=0.891, PV=1.05749E-12, TP=62, FP=7, Pos=166, Neg=173
S3,S4 7. IF(CDC42EP3 = 9.974..inf)) THEN (@Class = CONTROL)
   CF=0.954, PV=4.44089E-16, TP=63, FP=2, Pos=173, Neg=166
S1,S4 11. IF(ERG = 7.839..inf)) THEN (@Class = CASE)
   CF=0.944, PV=1.82365E-12, TP=61, FP=2, Pos=167, Neg=124
S2,S4 7. IF(CYP3A5 = 8.056..inf)) THEN (@Class = CONTROL)
   CF=0.956, PV=3.19411E-12, TP=40, FP=1, Pos=118, Neg=134
S1,S4 20. IF(DNAJC15 = -inf..7.955)) THEN (@Class = CASE)
   CF=0.827, PV=9.58703E-10, TP=81, FP=12, Pos=167, Neg=124
S1,S3 9. IF(ATP11B = 8.852..inf)) THEN (@Class = CASE)
   CF=0.957, PV=1.85698E-11, TP=50, FP=1, Pos=196, Neg=153
```

APPENDIX F

ITRL - SUPPLEMENTARY

F.1 CLASSIFICATION PERFORMANCE

Table 81: Classification performance of iTRL Combo, with two sources on GEO16011, brain cancer

Source	Target	SN	SP	BACC	AccAb
GEO4271_GEO4290	GEO16011	98.113	0.000	49.057	89.143
GEO4290_GEO4412	GEO16011	98.734	0.000	49.367	89.143
GEO4412_GEO4290	GEO16011	98.734	0.000	49.367	89.143
GEO4412_GEO4271	GEO16011	96.855	6.250	51.553	88.571
GEO4290_GEO4271	GEO16011	96.855	6.250	51.553	88.571
GEO4271_GEO4412	GEO16011	97.484	6.250	51.867	89.143
GEO1993_GEO4290	GEO16011	98.734	6.667	52.700	89.714
GEO1993_GEO4412	GEO16011	98.742	6.667	52.704	90.286
GEO1993_GEO4271	GEO16011	95.597	12.500	54.049	88.000
GEO4271_GEO1993	GEO16011	97.484	12.500	54.992	89.714
GEO4412_GEO1993	GEO16011	98.113	12.500	55.307	90.286
GEO4290_GEO1993	GEO16011	98.742	12.500	55.621	90.857

Table 82: Classification performance of iTRL Combo, with three sources on GEO16011, brain cancer

Source	Target	SN	SP	BACC	AccAb
GEO4412_GEO4271_GEO4290	GEO16011	98.113	0.000	49.057	89.143
GEO4271_GEO4412_GEO4290	GEO16011	98.734	0.000	49.367	89.143
GEO4271_GEO4290_GEO4412	GEO16011	98.734	0.000	49.367	89.143
GEO4412_GEO1993_GEO4271	GEO16011	95.597	6.250	50.924	87.429
GEO4290_GEO4412_GEO4271	GEO16011	96.855	6.250	51.553	88.571
GEO4412_GEO4290_GEO4271	GEO16011	96.855	6.250	51.553	88.571
GEO1993_GEO4271_GEO4412	GEO16011	97.484	6.250	51.867	89.143
GEO4290_GEO4271_GEO4412	GEO16011	97.484	6.250	51.867	89.143
GEO1993_GEO4412_GEO4271	GEO16011	97.484	6.250	51.867	89.143
GEO4271_GEO1993_GEO4412	GEO16011	98.113	6.250	52.182	89.714
GEO1993_GEO4271_GEO4290	GEO16011	98.113	6.250	52.182	89.714
GEO1993_GEO4290_GEO4412	GEO16011	98.734	6.667	52.700	89.714
GEO4412_GEO1993_GEO4290	GEO16011	98.734	6.667	52.700	89.714
GEO1993_GEO4412_GEO4290	GEO16011	98.734	6.667	52.700	89.714
GEO4290_GEO1993_GEO4412	GEO16011	98.742	6.667	52.704	90.286
GEO1993_GEO4290_GEO4271	GEO16011	95.597	12.500	54.049	88.000
GEO4290_GEO1993_GEO4271	GEO16011	95.597	12.500	54.049	88.000
GEO4412_GEO4271_GEO1993	GEO16011	97.484	12.500	54.992	89.714
GEO4290_GEO4271_GEO1993	GEO16011	98.113	12.500	55.307	90.286
GEO4271_GEO4412_GEO1993	GEO16011	98.113	12.500	55.307	90.286
GEO4271_GEO1993_GEO4290	GEO16011	98.113	12.500	55.307	90.286
GEO4290_GEO4412_GEO1993	GEO16011	98.742	12.500	55.621	90.857
GEO4412_GEO4290_GEO1993	GEO16011	98.742	12.500	55.621	90.857
GEO4271_GEO4290_GEO1993	GEO16011	98.742	12.500	55.621	90.857

Table 83: Classification performance of iTRL OnlyPriors, with two sources on GEO16011, brain cancer

Source	Target	SN	SP	BACC	AccAb
GEO4412_GEO1993	GEO16011	97.345	0.00	48.67	62.86
GEO4290_GEO4412	GEO16011	98.333	0.00	49.17	67.43
GEO4412_GEO4290	GEO16011	98.438	0.00	49.22	72.00
GEO4412_GEO4271	GEO16011	98.496	0.00	49.25	74.86
GEO4290_GEO4271	GEO16011	100.000	0.00	50.00	60.00
GEO4271_GEO4290	GEO16011	94.964	20.00	57.48	76.00
GEO4290_GEO1993	GEO16011	98.276	25.00	61.64	65.71
GEO4271_GEO1993	GEO16011	92.199	37.50	64.85	76.00
GEO1993_GEO4290	GEO16011	94.175	50.00	72.09	56.57
GEO1993_GEO4412	GEO16011	96.774	50.00	73.39	71.43
GEO4271_GEO4412	GEO16011	91.367	55.56	73.46	75.43
GEO1993_GEO4271	GEO16011	94.245	60.00	77.12	78.29

Table 84: Classification performance of iTRL OnlyPriors, with three sources on GEO16011, brain cancer

Source	Target	SN	SP	BACC	AccAb
GEO4271_GEO4290_GEO1993	GEO16011	95.00	0.00	47.50	76.00
GEO4271_GEO4412_GEO4290	GEO16011	96.95	0.00	48.47	72.57
GEO4271_GEO4290_GEO4412	GEO16011	97.06	0.00	48.53	75.43
GEO4412_GEO1993_GEO4290	GEO16011	97.20	0.00	48.60	59.43
GEO4412_GEO1993_GEO4271	GEO16011	97.27	0.00	48.64	61.14
GEO4412_GEO4290_GEO1993	GEO16011	97.92	0.00	48.96	53.71
GEO4290_GEO4412_GEO1993	GEO16011	99.04	0.00	49.52	58.86
GEO4290_GEO1993_GEO4271	GEO16011	99.07	0.00	49.53	60.57
GEO4412_GEO4271_GEO4290	GEO16011	99.20	0.00	49.60	70.86
GEO4412_GEO4290_GEO4271	GEO16011	99.20	0.00	49.60	70.86
GEO4412_GEO4271_GEO1993	GEO16011	99.21	0.00	49.60	71.43
GEO4290_GEO4412_GEO4271	GEO16011	100.00	0.00	50.00	64.00
GEO4290_GEO4271_GEO4412	GEO16011	100.00	0.00	50.00	58.86
GEO4290_GEO4271_GEO1993	GEO16011	100.00	0.00	50.00	57.14
GEO4271_GEO1993_GEO4412	GEO16011	91.97	33.33	62.65	73.71
GEO4271_GEO4412_GEO1993	GEO16011	90.71	40.00	65.36	74.86
GEO4290_GEO1993_GEO4412	GEO16011	98.21	33.33	65.77	63.43
GEO1993_GEO4412_GEO4290	GEO16011	99.07	33.33	66.20	61.71
GEO1993_GEO4271_GEO4412	GEO16011	94.44	44.44	69.44	60.57
GEO4271_GEO1993_GEO4290	GEO16011	96.90	42.86	69.88	73.14
GEO1993_GEO4290_GEO4271	GEO16011	94.95	50.00	72.48	54.86
GEO1993_GEO4290_GEO4412	GEO16011	96.88	50.00	73.44	54.29
GEO1993_GEO4271_GEO4290	GEO16011	92.55	57.14	74.85	52.00
GEO1993_GEO4412_GEO4271	GEO16011	96.40	55.56	75.98	64.00

Table 85: The average BACC per number of sources for iTRL-Combo

Target	**RL**	**TRL**	**iTRL_2**	**iTRL_3**	**iTRL_4**
GEO16011	49.367	51.399	52.345	52.853	52.797
GEO1993	82.996	82.642	82.973	82.979	83.092
GEO4271	54.167	58.333	60.316	60.842	61.248
GEO4290	55.263	61.923	63.770	64.222	64.674
GEO4412	74.337	74.438	76.643	77.204	77.852
GEO10780	71.250	72.370	72.322	73.228	72.022
GEO15852	84.884	85.466	85.465	85.562	85.853
GEO29431	100.000	91.551	90.278	89.661	89.815
GEO42568	90.625	89.960	89.979	90.039	89.989
GEO7904	70.113	69.601	69.289	69.023	69.023
GEO10715	60.417	69.357	70.201	69.046	68.726
GEO20916	98.529	98.540	97.968	97.798	97.859
GEO23878	100.000	97.298	97.031	96.953	96.519
GEO24514	90.000	94.167	93.399	94.555	94.861
GEO9348	91.667	93.750	96.875	98.437	98.785
GEO10072	97.917	96.923	96.253	96.007	95.846
GEO18842	95.238	97.784	97.319	97.271	97.362
GEO19188	94.505	95.275	95.131	94.963	94.821
GEO19804	94.958	94.993	94.514	94.653	94.931
GEO7670	91.186	93.875	94.195	94.201	94.510
GEO4412	74.337	74.833	74.992	74.878	75.039
GEO6956	76.765	75.664	75.670	75.642	75.642
GEO7904	91.186	88.268	88.770	88.859	88.396
GEO7670	70.113	67.335	66.580	66.690	66.032
GEO9348	91.667	89.584	88.889	88.194	88.889
GEO17951	86.383	87.079	87.028	87.418	87.701
GEO32448	93.611	91.513	91.135	90.879	90.829
GEO46602	94.444	90.377	88.731	88.079	87.884
GEO6956	76.765	73.813	73.336	72.234	72.483
GEO82188	89.346	90.914	90.780	90.714	90.509

Table 86: The average AccAb per number of sources for iTRL-Combo

Target	**RL**	**TRL**	**iTRL_2**	**iTRL_3**	**iTRL_4**
GEO16011	89.143	89.286	89.381	89.476	89.476
GEO1993	86.207	86.638	86.782	86.638	86.638
GEO4271	78.000	80.000	80.667	80.833	81.000
GEO4290	82.000	83.750	84.000	83.917	83.833
GEO4412	81.176	82.059	83.627	83.971	84.559
GEO10780	86.486	87.162	87.162	87.628	87.027
GEO15852	84.884	85.466	85.465	85.562	85.853
GEO29431	93.939	95.076	95.329	95.392	95.644
GEO42568	95.868	96.281	96.556	96.660	96.591
GEO7904	77.419	77.822	77.688	77.553	77.553
GEO10715	60.000	67.500	68.611	68.195	68.333
GEO20916	97.143	97.857	97.738	97.738	97.738
GEO23878	94.915	95.763	96.751	96.963	96.681
GEO24514	93.878	96.428	95.578	96.428	96.854
GEO9348	95.122	97.561	98.679	99.339	99.644
GEO10072	96.262	96.729	96.262	96.067	95.950
GEO18842	91.209	97.527	97.069	97.115	97.207
GEO19188	94.872	95.513	95.393	95.246	95.112
GEO19804	94.167	94.792	94.514	94.653	94.931
GEO7670	87.879	93.940	94.318	94.381	94.634
GEO4412	81.176	82.059	82.059	82.059	82.157
GEO6956	87.640	86.236	86.423	86.470	86.470
GEO7904	87.879	89.053	89.647	89.836	89.457
GEO7670	77.419	78.629	78.091	78.158	77.755
GEO9348	95.122	96.951	96.748	96.545	96.748
GEO17951	83.212	85.949	86.496	87.044	87.348
GEO32448	88.750	90.313	90.625	90.573	90.677
GEO46602	92.000	94.000	93.167	92.917	92.750
GEO6956	87.640	86.236	85.955	85.393	85.206
GEO82188	87.500	90.809	90.747	90.686	90.472

Table 87: The average rate (%) of abstentions per number of sources for iTRL-Combo

Target	**RL**	**TRL**	**iTRL_2**	**iTRL_3**	**iTRL_4**
GEO16011	1.143	0.686	0.381	0.286	0.167
GEO1993	0.000	0.000	0.000	0.000	0.000
GEO4271	0.000	0.000	0.000	0.000	0.000
GEO4290	1.000	0.250	0.000	0.000	0.000
GEO4412	1.176	0.000	0.000	0.000	0.000
GEO10780	1.081	0.676	0.631	0.481	0.631
GEO15852	0.000	0.000	0.000	0.000	0.000
GEO29431	6.061	0.379	0.253	0.063	0.063
GEO42568	1.653	0.207	0.069	0.069	0.103
GEO7904	1.613	0.403	0.269	0.202	0.202
GEO10715	13.330	7.450	5.278	4.167	4.028
GEO20916	1.429	0.715	0.238	0.060	0.119
GEO23878	5.085	1.695	0.565	0.212	0.141
GEO24514	0.000	0.000	0.000	0.000	0.000
GEO9348	2.439	0.610	0.407	0.203	0.000
GEO10072	1.869	0.468	0.312	0.273	0.234
GEO18842	4.396	0.274	0.275	0.183	0.183
GEO19188	0.000	0.000	0.000	0.000	0.000
GEO19804	0.833	0.208	0.000	0.000	0.000
GEO7670	4.545	0.000	0.000	0.000	0.000
GEO4412	1.176	0.588	0.686	0.588	0.588
GEO6956	1.124	0.843	0.656	0.562	0.562
GEO7904	1.613	0.000	0.126	0.000	0.000
GEO7670	4.545	0.379	0.000	0.000	0.000
GEO9348	2.439	0.000	0.000	0.000	0.000
GEO17951	3.650	1.278	0.608	0.426	0.395
GEO32448	5.000	1.250	0.521	0.313	0.156
GEO46602	4.000	0.000	0.167	0.000	0.083
GEO6956	1.124	0.000	0.000	0.000	0.000
GEO82188	2.206	0.184	0.061	0.031	0.031

Table 88: The average BACC per number of sources for iTRL-OnlyPriors

Target	**RL**	**TRL**	**iTRL_2**	**iTRL_3**	**iTRL_4**
GEO16011	49.367	63.610	60.528	57.691	56.849
GEO1993	82.996	71.011	72.944	73.550	73.651
GEO4271	54.167	72.283	78.168	77.179	76.959
GEO4290	55.263	67.047	69.979	73.533	74.820
GEO4412	74.337	68.812	72.219	73.850	76.471
GEO10780	71.250	70.622	57.557	43.250	57.650
GEO15852	84.884	81.860	73.723	61.286	60.243
GEO29431	100.000	83.007	82.037	73.915	64.469
GEO42568	90.625	86.749	70.305	65.135	62.444
GEO7904	70.113	53.177	49.508	50.501	33.681
GEO10715	60.417	71.075	66.439	62.531	60.567
GEO20916	98.529	91.510	76.236	67.343	60.786
GEO23878	100.000	88.296	81.229	71.530	62.238
GEO24514	90.000	78.563	78.885	75.832	70.713
GEO9348	91.667	89.368	93.249	92.636	87.084
GEO10072	97.917	92.719	91.097	90.698	91.168
GEO18842	95.238	96.737	97.317	97.594	97.690
GEO19188	94.505	92.940	83.219	83.176	83.172
GEO19804	94.958	89.053	88.388	87.728	87.216
GEO7670	91.186	92.997	91.382	92.415	92.198
GEO4412	74.337	66.010	58.538	55.277	42.360
GEO6956	76.765	71.408	64.134	61.898	54.463
GEO7904	91.186	85.635	68.630	57.120	43.750
GEO7670	70.113	61.924	62.624	61.249	62.205
GEO9348	91.667	84.362	87.344	84.885	69.112
GEO17951	86.383	80.000	77.234	72.592	72.213
GEO32448	93.611	79.966	71.696	66.593	64.424
GEO46602	94.444	88.294	85.143	82.116	79.207
GEO6956	76.765	70.450	68.384	67.331	66.032
GEO82188	89.346	81.691	73.331	67.530	67.670

Table 89: The average AccAb per number of sources for iTRL-OnlyPriors

Target	**RL**	**TRL**	**iTRL_2**	**iTRL_3**	**iTRL_4**
GEO16011	89.143	78.286	69.714	64.310	61.762
GEO1993	86.207	74.569	67.960	60.920	56.394
GEO4271	78.000	80.250	64.583	52.167	45.875
GEO4290	82.000	74.250	67.917	61.208	53.083
GEO4412	81.176	73.824	65.588	53.676	43.235
GEO10780	86.486	31.622	10.180	5.203	7.432
GEO15852	84.884	63.663	41.473	23.546	13.324
GEO29431	93.939	82.576	59.596	36.427	17.992
GEO42568	95.868	84.504	45.799	25.999	9.332
GEO7904	77.419	39.516	19.758	9.812	2.218
GEO10715	60.000	35.000	21.111	19.167	17.778
GEO20916	97.143	74.999	42.976	36.964	32.381
GEO23878	94.915	73.305	50.848	33.051	20.410
GEO24514	93.878	78.565	46.429	36.395	29.507
GEO9348	95.122	87.195	57.825	37.246	24.035
GEO10072	96.262	85.981	82.399	80.880	80.724
GEO18842	91.209	92.857	91.209	90.064	89.148
GEO19188	94.872	90.705	80.396	78.419	76.336
GEO19804	94.167	82.500	77.778	76.840	75.868
GEO7670	87.879	90.909	80.808	74.684	70.897
GEO4412	81.176	52.647	27.451	16.373	9.510
GEO6956	87.640	32.023	15.471	10.768	6.835
GEO7904	87.879	71.970	36.111	19.697	11.995
GEO7670	77.419	48.791	28.898	21.371	16.264
GEO9348	95.122	93.902	49.593	28.760	16.718
GEO17951	83.212	67.336	53.285	42.245	35.462
GEO32448	88.750	65.313	52.708	47.500	42.135
GEO46602	92.000	91.500	86.000	79.500	75.083
GEO6956	87.640	58.708	54.682	48.830	43.258
GEO82188	87.500	69.669	51.777	41.146	33.977

Table 90: The average rate (%) of abstentions per number of sources for iTRL-OnlyPriors

Target	**RL**	**TRL**	**iTRL_2**	**iTRL_3**	**iTRL_4**
GEO16011	1.143	15.714	25.619	31.976	35.286
GEO1993	0.000	5.604	18.103	28.089	34.483
GEO4271	0.000	4.250	24.167	37.708	45.250
GEO4290	1.000	7.750	17.000	26.167	35.292
GEO4412	1.176	7.941	21.079	35.098	47.794
GEO10780	1.081	59.189	87.027	93.536	89.302
GEO15852	0.000	24.709	50.581	71.899	84.206
GEO29431	6.061	8.712	34.848	60.038	79.798
GEO42568	1.653	8.265	49.931	71.660	89.842
GEO7904	1.613	39.919	70.296	85.349	95.631
GEO10715	13.333	54.167	71.111	70.139	70.278
GEO20916	1.429	18.214	51.667	57.798	62.500
GEO23878	5.085	19.068	44.350	62.994	76.695
GEO24514	0.000	30.102	47.449	59.099	66.837
GEO9348	2.439	8.232	39.634	61.230	74.797
GEO10072	1.869	7.710	12.461	14.759	15.343
GEO18842	4.396	4.121	6.410	7.876	8.883
GEO19188	0.000	2.885	14.263	16.346	18.643
GEO19804	0.833	7.500	13.472	14.618	15.451
GEO7670	4.545	2.273	12.752	19.949	23.737
GEO4412	1.176	32.647	64.608	79.510	87.843
GEO6956	1.124	57.584	79.084	86.096	91.292
GEO7904	1.613	35.484	56.439	75.821	84.407
GEO7670	4.545	18.182	61.694	73.992	81.116
GEO9348	2.439	1.525	45.935	68.750	82.165
GEO17951	3.650	15.694	32.360	46.320	55.839
GEO32448	5.000	20.000	29.688	35.313	42.917
GEO46602	4.000	1.500	6.333	12.583	16.917
GEO6956	1.124	22.472	28.558	36.985	44.335
GEO82188	2.206	16.177	35.662	48.101	57.874

Table 91: Accuracy of iTRL with two best sources

Source	**Target**	**SN**	**SP**	**BACC**	**AccAb**
GEO4290_GEO1993	GEO16011	98.742	12.500	55.621	90.857
GEO4271_GEO16011	GEO1993	97.436	73.684	85.560	89.655
GEO4290_GEO4412	GEO4271	100.000	25.000	62.500	82.000
GEO4271_GEO16011	GEO4290	97.531	36.842	67.186	86.000
GEO1993_GEO16011	GEO4412	93.220	69.231	81.226	85.882
GEO15852_GEO42568	GEO10780	50.000	100.000	75.000	88.649
GEO7904_GEO10780	GEO15852	90.698	83.721	87.209	87.209
GEO7904_GEO10780	GEO29431	96.296	100.000	98.148	95.455
GEO15852_GEO10780	GEO42568	96.154	94.118	95.136	95.868
GEO29431_GEO15852	GEO7904	90.698	52.632	71.665	79.032
GEO23878_GEO20916	GEO10715	83.333	70.000	76.667	73.333
GEO10715_GEO23878	GEO20916	100.000	100.000	100.000	100.000
GEO20916_GEO24514	GEO23878	100.000	100.000	100.000	100.000
GEO10715_GEO20916	GEO24514	100.000	93.333	96.667	97.959
GEO10715_GEO20916	GEO9348	100.000	100.000	100.000	100.000
GEO19188_GEO19804	GEO10072	100.000	95.918	97.959	98.131
GEO19804_GEO10072	GEO18842	97.826	100.000	98.913	98.901
GEO18842_GEO10072	GEO19188	96.703	93.846	95.275	95.513
GEO19188_GEO7670	GEO19804	96.667	95.000	95.833	95.833
GEO19804_GEO18842	GEO7670	94.872	100.000	97.436	96.970
GEO9348_GEO6956	GEO4412	98.305	57.692	77.999	85.882
GEO9348_GEO7904	GEO6956	98.551	55.000	76.775	88.764
GEO4412_GEO9348	GEO7670	94.872	92.308	93.590	92.424
GEO7670_GEO4412	GEO7904	100.000	36.842	68.421	80.645
GEO6956_GEO7904	GEO9348	100.000	91.667	95.833	98.780
GEO82188_GEO6956	GEO17951	89.706	86.957	88.331	88.321
GEO46602_GEO17951	GEO32448	97.500	92.500	95.000	95.000
GEO17951_GEO32448	GEO46602	100.000	92.857	96.429	98.000
GEO32448_GEO46602	GEO6956	100.000	55.000	77.500	89.888
GEO46602_GEO32448	GEO82188	90.769	95.775	93.272	93.382

Table 92: Accuracy of iTRL with 3 best sources

Source	Target	SN	SP	BACC	AccAb
GEO4271_GEO4290_GEO1993	GEO16011	98.742	12.500	55.621	90.857
GEO4271_GEO4290_GEO16011	GEO1993	97.436	73.684	85.560	89.655
GEO16011_GEO4412_GEO1993	GEO4271	98.684	29.167	63.925	82.000
GEO4271_GEO16011_GEO1993	GEO4290	98.765	36.842	67.804	87.000
GEO4271_GEO1993_GEO16011	GEO4412	93.220	69.231	81.226	85.882
GEO15852_GEO29431_GEO42568	GEO10780	50.000	100.000	75.000	88.649
GEO7904_GEO10780_GEO29431	GEO15852	90.698	86.047	88.372	88.372
GEO42568_GEO7904_GEO10780	GEO29431	94.444	100.000	97.222	93.939
GEO29431_GEO15852_GEO10780	GEO42568	96.154	94.118	95.136	95.868
GEO42568_GEO29431_GEO15852	GEO7904	90.698	52.632	71.665	79.032
GEO24514_GEO23878_GEO9348	GEO10715	78.947	72.727	75.837	76.667
GEO24514_GEO10715_GEO23878	GEO20916	100.000	100.000	100.000	100.000
GEO9348_GEO20916_GEO24514	GEO23878	100.000	100.000	100.000	100.000
GEO10715_GEO23878_GEO20916	GEO24514	100.000	93.333	96.667	97.959
GEO20916_GEO24514_GEO10715	GEO9348	100.000	100.000	100.000	100.000
GEO18842_GEO19188_GEO19804	GEO10072	100.000	95.918	97.959	98.131
GEO7670_GEO19804_GEO10072	GEO18842	100.000	100.000	100.000	100.000
GEO18842_GEO19804_GEO10072	GEO19188	96.703	93.846	95.275	95.513
GEO18842_GEO19188_GEO7670	GEO19804	96.667	95.000	95.833	95.833
GEO19188_GEO19804_GEO18842	GEO7670	97.436	96.296	96.866	96.970
GEO7670_GEO9348_GEO6956	GEO4412	98.305	65.385	81.845	88.235
GEO9348_GEO4412_GEO7904	GEO6956	98.551	55.000	76.775	88.764
GEO7904_GEO9348_GEO4412	GEO7670	92.308	92.593	92.450	92.424
GEO4412_GEO6956_GEO7670	GEO7904	97.674	42.105	69.890	80.645
GEO4412_GEO7670_GEO7904	GEO9348	100.000	91.667	95.833	98.780
GEO32448_GEO82188_GEO6956	GEO17951	92.647	86.957	89.802	89.781
GEO6956_GEO46602_GEO17951	GEO32448	97.500	92.500	95.000	95.000
GEO32448_GEO17951_GEO82188	GEO46602	100.000	92.857	96.429	98.000
GEO17951_GEO32448_GEO46602	GEO6956	100.000	55.000	77.500	89.888
GEO46602_GEO6956_GEO32448	GEO82188	92.308	94.366	93.337	93.382

Table 93: Accuracy of iTRL with four best sources

Source	Target	SN	SP	BACC	AccAb
GEO4412_GEO4271_GEO4290_GEO1993	GEO16011	98.74	12.50	55.62	90.86
GEO4271_GEO16011_GEO4412_GEO4290	GEO1993	97.44	73.68	85.56	89.66
GEO4290_GEO16011_GEO4412_GEO1993	GEO4271	98.68	29.17	63.93	82.00
GEO16011_GEO4412_GEO4271_GEO1993	GEO4290	96.30	36.84	66.57	85.00
GEO4271_GEO4290_GEO1993_GEO16011	GEO4412	93.22	69.23	81.23	85.88
GEO29431_GEO7904_GEO15852_GEO42568	GEO10780	47.62	100.00	73.81	88.11
GEO42568_GEO7904_GEO10780_GEO29431	GEO15852	90.70	86.05	88.37	88.37
GEO15852_GEO42568_GEO7904_GEO10780	GEO29431	94.44	100.00	97.22	93.94
GEO15852_GEO29431_GEO7904_GEO10780	GEO42568	99.04	88.24	93.64	97.52
GEO42568_GEO10780_GEO29431_GEO15852	GEO7904	90.70	52.63	71.67	79.03
GEO20916_GEO24514_GEO23878_GEO9348	GEO10715	84.21	63.64	73.92	76.67
GEO9348_GEO24514_GEO10715_GEO23878	GEO20916	100.00	100.00	100.00	100.00
GEO9348_GEO20916_GEO10715_GEO24514	GEO23878	100.00	95.83	97.92	98.31
GEO20916_GEO23878_GEO10715_GEO9348	GEO24514	100.00	93.33	96.67	97.96
GEO20916_GEO23878_GEO10715_GEO24514	GEO9348	100.00	100.00	100.00	100.00
GEO18842_GEO19188_GEO7670_GEO19804	GEO10072	100.00	93.88	96.94	97.20
GEO7670_GEO19188_GEO19804_GEO10072	GEO18842	100.00	100.00	100.00	100.00
GEO18842_GEO19804_GEO10072_GEO7670	GEO19188	96.70	93.85	95.28	95.51
GEO18842_GEO10072_GEO19188_GEO7670	GEO19804	96.67	95.00	95.83	95.83
GEO10072_GEO19188_GEO19804_GEO18842	GEO7670	97.44	96.30	96.87	96.97
GEO7670_GEO9348_GEO7904_GEO6956	GEO4412	98.31	65.39	81.85	88.24
GEO7670_GEO4412_GEO9348_GEO7904	GEO6956	98.55	55.00	76.78	88.76
GEO4412_GEO6956_GEO9348_GEO7904	GEO7670	94.87	88.89	91.88	92.42
GEO9348_GEO4412_GEO6956_GEO7670	GEO7904	97.67	42.11	69.89	80.65
GEO6956_GEO4412_GEO7670_GEO7904	GEO9348	100.00	91.67	95.83	98.78
GEO46602_GEO32448_GEO82188_GEO6956	GEO17951	92.65	86.96	89.80	89.78
GEO82188_GEO17951_GEO6956_GEO46602	GEO32448	100.00	90.00	95.00	95.00
GEO17951_GEO82188_GEO6956_GEO32448	GEO46602	100.00	92.86	96.43	98.00
GEO82188_GEO17951_GEO46602_GEO32448	GEO6956	98.55	55.00	76.78	88.76
GEO17951_GEO46602_GEO32448_GEO6956	GEO82188	95.39	88.73	92.06	91.91

F.2 ROBUST RULE PATTERNS VIA ITRL

F.2.1 BRAIN CANCER

```
 === TARGET ===
GEO16011.txt

=== SOURCES ===
S1: GEO1993.txt
S2: GEO4271.txt
S3: GEO4290.txt
S4: GEO4412.txt

=== Robust Rules ===
S1 6. ((ARPC1A = -inf..10.395) (MYL9 = -inf..5.967)) ==> (@Class = CONTROL)
   CF=0.861, PV=3.83027E-14, TP=40, FP=22, Pos=102, Neg=398
S1 5. ((HSP90B1 = 10.767..inf) (PLOD1 = 8.723..inf)) ==> (@Class = CASE)
   CF=0.913, PV=1.49425E-12, TP=174, FP=4, Pos=398, Neg=102
S1 8. ((HSP90B1 = 10.767..inf) (GLUD1 = -inf..10.090)) ==> (@Class = CASE)
   CF=0.887, PV=2.25053E-12, TP=191, FP=6, Pos=398, Neg=102
S3 9. ((LAMC1 = -inf..8.484) (MYL9 = 5.967..inf)
             (SYPL1 = 9.243..inf)) ==> (@Class = CONTROL)
   CF=0.812, PV=3.96806E-8, TP=25, FP=24, Pos=59, Neg=283
S1 5. ((MYL9 = -inf..5.664) (C5orf13 = -inf..8.444)) ==> (@Class = CONTROL)
   CF=0.926, PV=2.0407E-12, TP=21, FP=3, Pos=102, Neg=398
S2 3. ((LDHA = 12.680..inf)) ==> (@Class = CASE)
   CF=0.933, PV=1.00142E-13, TP=192, FP=3, Pos=359, Neg=84
S1 5. ((CALU = 8.784..inf)) ==> (@Class = CASE)
   CF=0.871, PV=5.70211E-13, TP=214, FP=8, Pos=398, Neg=103
S3 1. ((HSPA9 = 10.061..inf) (CALU = 8.784..inf)) ==> (@Class = CASE)
```

```
   CF=0.951, PV=4.53005E-8, TP=111, FP=1, Pos=283, Neg=60
S3 2. ((PLOD1 = 8.718..inf) (ARPC1A = 10.395..inf)) ==> (@Class = CASE)
   CF=0.936, PV=3.62683E-10, TP=151, FP=2, Pos=283, Neg=60
S3 3. ((EPAS1 = 9.104..inf) (ANXA7 = -inf..10.059)) ==> (@Class = CASE)
   CF=0.949, PV=1.25205E-7, TP=107, FP=1, Pos=283, Neg=59

=== TARGET ===
GEO4412.txt

=== SOURCES ===
S1: GEO1993.txt
S2: GEO4271.txt
S3: GEO4290.txt
S4: GEO16011.txt

=== Robust Rules ===
S3 9. ((NUCB1 = 12.539..inf)) ==> (@Class = CASE)
   CF=0.879, PV=3.49414E-6, TP=116, FP=3, Pos=293, Neg=58
S3 10. ((P4HB = 14.225..inf)) ==> (@Class = CASE)
   CF=0.874, PV=7.49962E-6, TP=111, FP=3, Pos=293, Neg=58
S2 11. ((PPP2R5A = 12.491..inf)) ==> (@Class = CONTROL)
   CF=0.872, PV=1.65312E-13, TP=33, FP=18, Pos=82, Neg=369
S1 12. ((ABLIM1 = 14.070..inf)) ==> (@Class = CONTROL)
   CF=0.869, PV=4.27436E-14, TP=37, FP=19, Pos=101, Neg=408
S2 1. ((CALU = 13.234..inf) (WDR77 = 11.469..inf)) ==> (@Class = CASE)
   CF=0.915, PV=4.607E-10, TP=155, FP=3, Pos=369, Neg=82
S2 6. ((PPP2R5A = 12.456..inf)) ==> (@Class = CONTROL)
   CF=0.87, PV=1.05804E-13, TP=34, FP=19, Pos=82, Neg=369
S1 12. ((IGFBP2 = -inf..11.418)) ==> (@Class = CONTROL)
   CF=0.842, PV=0E0, TP=57, FP=40, Pos=101, Neg=408
```

```
S3 10. ((P4HB = 14.225..inf) (PLOD3 = 12.986..inf)) ==> (@Class = CASE)
   CF=0.926, PV=1.82973E-8, TP=135, FP=2, Pos=293, Neg=59
S1 17. ((PPP2R5A = 12.456..inf)) ==> (@Class = CONTROL)
   CF=0.882, PV=1.11022E-16, TP=42, FP=19, Pos=102, Neg=408
S1 18. ((ANXA5 = -inf..13.856) (CDC20 = -inf..9.246)) ==> (@Class = CONTROL)
   CF=0.854, PV=6.32161E-13, TP=37, FP=22, Pos=102, Neg=408
S1 9. ((SMC4 = -inf..11.376)) ==> (@Class = CONTROL)
   CF=0.867, PV=5.55112E-16, TP=44, FP=24, Pos=101, Neg=409

=== TARGET ===
GEO1993.txt

=== SOURCES ===
S1: GEO4271.txt
S2: GEO4290.txt
S3: GEO16011.txt
S4: GEO4412.txt

=== Robust rules ===
S1 7. ((GSS = -inf..7.641) (ARL4C = -inf..7.900)) ==> (@Class = CONTROL)
   CF=0.892, PV=1.60094E-13, TP=30, FP=11, Pos=102, Neg=410
S1 8. ((GLUD1 = 8.976..inf) (MYL9 = -inf..6.190)) ==> (@Class = CONTROL)
   CF=0.886, PV=2.02061E-14, TP=34, FP=14, Pos=102, Neg=410
S1 6. ((SMC4 = -inf..6.228)) ==> (@Class = CONTROL)
   CF=0.892, PV=1.55431E-15, TP=36, FP=14, Pos=102, Neg=410
S1 7. ((PRKCZ = -inf..6.802)) ==> (@Class = CASE)
   CF=0.888, PV=6.03522E-7, TP=103, FP=3, Pos=410, Neg=102
S2 6. ((PFN1 = 10.507..inf)) ==> (@Class = CASE)
   CF=0.886, PV=8.97637E-7, TP=107, FP=3, Pos=334, Neg=78
S2 9. ((PKM2 = 10.472..inf) (PTPN12 = 6.660..inf)) ==> (@Class = CASE)
```

```
   CF=0.885, PV=1.61344E-8, TP=139, FP=4, Pos=334, Neg=78
S2 4. ((ANXA7 = -inf..8.470) (ATF3 = 6.741..inf)) ==> (@Class = CASE)
   CF=0.95, PV=9.78106E-14, TP=168, FP=2, Pos=329, Neg=83
S2 6. ((COL3A1 = -inf..7.259)) ==> (@Class = CONTROL)
   CF=0.907, PV=7.88258E-15, TP=31, FP=9, Pos=84, Neg=329
S1 9. ((ADD3 = -inf..9.980) (PGM1 = 8.495..inf)) ==> (@Class = CASE)
   CF=0.876, PV=2.55884E-12, TP=203, FP=7, Pos=410, Neg=103
S1 10. ((ABLIM1 = 9.413..inf)) ==> (@Class = CONTROL)
   CF=0.885, PV=1.34004E-13, TP=32, FP=13, Pos=83, Neg=329
S2 5. ((GLUD1 = 8.976..inf) (MGP = -inf..7.231)) ==> (@Class = CONTROL)
   CF=0.917, PV=5.89528E-14, TP=24, FP=6, Pos=57, Neg=270
S1 6. ((SMC4 = -inf..6.228)) ==> (@Class = CONTROL)
   CF=0.914, PV=1.66533E-15, TP=31, FP=8, Pos=83, Neg=329
S3 7. ((MTHFD2 = -inf..7.284)) ==> (@Class = CASE)
   CF=0.851, PV=2.37646E-3, TP=72, FP=2, Pos=194, Neg=33
S1 8. ((MGP = -inf..6.907)) ==> (@Class = CONTROL)
   CF=0.849, PV=1.73174E-11, TP=33, FP=20, Pos=83, Neg=329
S2 8. ((GLUD1 = 8.976..inf)) ==> (@Class = CONTROL)
   CF=0.876, PV=4.36162E-12, TP=27, FP=14, Pos=57, Neg=270

=== TARGET ===
GEO4271.txt

=== SOURCES ===
S1: GEO4412.txt
S2: GEO1993.txt
S3: GEO4290.txt
S4: GEO16011.txt

=== Robust Rules ===
```

```
S1 5. ((GLUD1 = -inf..10.960) (CCT6A = 10.397..inf)) ==> (@Class = CASE)
   CF=0.928, PV=8.69818E-9, TP=115, FP=2, Pos=407, Neg=101
```

F.2.2 BREAST CANCER

```
 === TARGET ===
GEO29431.txt

=== SOURCES ===
S1: GEO7904.txt
S2: GEO10780.txt
S3: GEO15852.txt
S4: GEO42568.txt

=== Robust Rules ===
S2 2. ((BNIP3L = 10.814..inf)) ==> (@Class = CONTROL)
   CF=0.977, PV=7.80487E-14, TP=42, FP=0, Pos=214, Neg=237
S3 5. ((ADH1C = 6.030..inf)) ==> (@Class = CONTROL)
   CF=0.962, PV=0E0, TP=43, FP=2, Pos=71, Neg=195
S4 6. ((ACACB = -inf..10.741)) ==> (@Class = CASE)
   CF=0.958, PV=1.73878E-9, TP=147, FP=1, Pos=152, Neg=28
S3 7. ((ACACB = 10.741..inf)) ==> (@Class = CONTROL)
   CF=0.944, PV=0E0, TP=59, FP=7, Pos=71, Neg=195
S3 2. ((AOC3 = 10.329..inf)) ==> (@Class = CONTROL)
   CF=0.96, PV=0E0, TP=49, FP=3, Pos=71, Neg=195
S3 3. ((ANGPT1 = 7.790..inf)) ==> (@Class = CONTROL)
   CF=0.969, PV=0E0, TP=30, FP=0, Pos=70, Neg=196
S3 4. ((CCT3 = 10.149..inf)) ==> (@Class = CASE)
   CF=0.967, PV=1.82521E-13, TP=112, FP=1, Pos=196, Neg=70
S3 7. ((AOC3 = -inf..8.790)) ==> (@Class = CASE)
```

```
   CF=0.902, PV=9.99201E-15, TP=160, FP=6, Pos=196, Neg=71
S3 2. ((ASPA = 8.321..inf)) ==> (@Class = CONTROL)
   CF=0.975, PV=0E0, TP=38, FP=0, Pos=71, Neg=196
S3 3. ((ADH1B = 11.084..inf)) ==> (@Class = CONTROL)
   CF=0.963, PV=0E0, TP=44, FP=2, Pos=71, Neg=196
S3 4. ((COL11A1 = 7.688..inf)) ==> (@Class = CASE)
   CF=0.938, PV=5.21805E-15, TP=140, FP=3, Pos=196, Neg=71
S3 2. ((ECT2 = 7.814..inf)) ==> (@Class = CASE)
   CF=0.973, PV=2.22045E-16, TP=134, FP=1, Pos=196, Neg=71
S3 6. ((CLDN5 = 9.204..inf)) ==> (@Class = CONTROL)
   CF=0.904, PV=3.21965E-15, TP=39, FP=9, Pos=71, Neg=196

=== TARGET ===
GEO42568.txt

=== SOURCES ===
S1: GEO7904.txt
S2: GEO10780.txt
S3: GEO29431.txt
S4: GEO15852.txt

=== Robust Rules ===
S3 2. ((GOS2 = 12.039..inf)) ==> (@Class = CONTROL)
   CF=0.968, PV=0E0, TP=40, FP=1, Pos=70, Neg=191
S2 1. ((CAP1 = 11.432..inf) (FHL1 = -inf..7.634)) ==> (@Class = CASE)
   CF=0.98, PV=0E0, TP=107, FP=1, Pos=239, Neg=200
S2 2. ((ADIPOQ = -inf..4.498)) ==> (@Class = CASE)
   CF=0.912, PV=1.07248E-13, TP=71, FP=5, Pos=238, Neg=201
S2 3. ((ETNK1 = 8.188..inf)) ==> (@Class = CASE)
   CF=0.882, PV=0E0, TP=122, FP=13, Pos=239, Neg=201
```

```
S1 4. ((ADIPOQ = 7.981..inf) (ADNP = -inf..7.712)) ==> (@Class = CONTROL)
   CF=0.884, PV=0E0, TP=132, FP=20, Pos=230, Neg=281

=== TARGET ===
GEO15852.txt

=== SOURCES ===
S1: GEO42568.txt
S2: GEO7904.txt
S3: GEO10780.txt
S4: GEO29431.txt

=== Robust Rules ===
S3 1. ((ACIN1 = 8.559..inf)) ==> (@Class = CONTROL)
   CF=0.938, PV=0E0, TP=122, FP=5, Pos=193, Neg=135
S1 4. ((ADAR = 9.355..inf)) ==> (@Class = CASE)
   CF=0.807, PV=7.9492E-13, TP=116, FP=22, Pos=282, Neg=230
S1 1. ((GPR157 = 7.558..inf)) ==> (@Class = CASE)
   CF=0.93, PV=1.22125E-15, TP=76, FP=4, Pos=276, Neg=232
```

F.2.3 COLON CANCER

```
 === TARGET ===
GEO29431.txt

=== SOURCES ===
S1: GEO7904.txt
S2: GEO10780.txt
S3: GEO15852.txt
S4: GEO42568.txt
```

```
=== Robust Rules ===
S2 2. ((BNIP3L = 10.814..inf)) ==> (@Class = CONTROL)
   CF=0.977, PV=7.80487E-14, TP=42, FP=0, Pos=214, Neg=237
S3 5. ((ADH1C = 6.030..inf)) ==> (@Class = CONTROL)
   CF=0.962, PV=0E0, TP=43, FP=2, Pos=71, Neg=195
S4 6. ((ACACB = -inf..10.741)) ==> (@Class = CASE)
   CF=0.958, PV=1.73878E-9, TP=147, FP=1, Pos=152, Neg=28
S3 7. ((ACACB = 10.741..inf)) ==> (@Class = CONTROL)
   CF=0.944, PV=0E0, TP=59, FP=7, Pos=71, Neg=195
S3 2. ((AOC3 = 10.329..inf)) ==> (@Class = CONTROL)
   CF=0.96, PV=0E0, TP=49, FP=3, Pos=71, Neg=195
S3 3. ((ANGPT1 = 7.790..inf)) ==> (@Class = CONTROL)
   CF=0.969, PV=0E0, TP=30, FP=0, Pos=70, Neg=196
S3 4. ((CCT3 = 10.149..inf)) ==> (@Class = CASE)
   CF=0.967, PV=1.82521E-13, TP=112, FP=1, Pos=196, Neg=70
S3 7. ((AOC3 = -inf..8.790)) ==> (@Class = CASE)
   CF=0.902, PV=9.99201E-15, TP=160, FP=6, Pos=196, Neg=71
S3 2. ((ASPA = 8.321..inf)) ==> (@Class = CONTROL)
   CF=0.975, PV=0E0, TP=38, FP=0, Pos=71, Neg=196
S3 3. ((ADH1B = 11.084..inf)) ==> (@Class = CONTROL)
   CF=0.963, PV=0E0, TP=44, FP=2, Pos=71, Neg=196
S3 4. ((COL11A1 = 7.688..inf)) ==> (@Class = CASE)
   CF=0.938, PV=5.21805E-15, TP=140, FP=3, Pos=196, Neg=71
S3 2. ((ECT2 = 7.814..inf)) ==> (@Class = CASE)
   CF=0.973, PV=2.22045E-16, TP=134, FP=1, Pos=196, Neg=71
S3 6. ((CLDN5 = 9.204..inf)) ==> (@Class = CONTROL)
   CF=0.904, PV=3.21965E-15, TP=39, FP=9, Pos=71, Neg=196

=== TARGET ===
```

GEO42568.txt

=== SOURCES ===

S1: GEO7904.txt

S2: GEO10780.txt

S3: GEO29431.txt

S4: GEO15852.txt

=== Robust Rules ===

S3 2. ((GOS2 = 12.039..inf)) ==> (@Class = CONTROL)

   CF=0.968, PV=0E0, TP=40, FP=1, Pos=70, Neg=191

S2 1. ((CAP1 = 11.432..inf) (FHL1 = -inf..7.634)) ==> (@Class = CASE)

   CF=0.98, PV=0E0, TP=107, FP=1, Pos=239, Neg=200

S2 2. ((ADIPOQ = -inf..4.498)) ==> (@Class = CASE)

   CF=0.912, PV=1.07248E-13, TP=71, FP=5, Pos=238, Neg=201

S2 3. ((ETNK1 = 8.188..inf)) ==> (@Class = CASE)

   CF=0.882, PV=0E0, TP=122, FP=13, Pos=239, Neg=201

S1 4. ((ADIPOQ = 7.981..inf) (ADNP = -inf..7.712)) ==> (@Class = CONTROL)

   CF=0.884, PV=0E0, TP=132, FP=20, Pos=230, Neg=281

=== TARGET ===

GEO15852.txt

=== SOURCES ===

S1: GEO42568.txt

S2: GEO7904.txt

S3: GEO10780.txt

S4: GEO29431.txt

=== Robust Rules ===

```
S3 1. ((ACIN1 = 8.559..inf)) ==> (@Class = CONTROL)
   CF=0.938, PV=0E0, TP=122, FP=5, Pos=193, Neg=135
S1 4. ((ADAR = 9.355..inf)) ==> (@Class = CASE)
   CF=0.807, PV=7.9492E-13, TP=116, FP=22, Pos=282, Neg=230
S1 1. ((GPR157 = 7.558..inf)) ==> (@Class = CASE)
   CF=0.93, PV=1.22125E-15, TP=76, FP=4, Pos=276, Neg=232
```

F.2.4 LUNG CANCER

```
=== TARGET ===
GEO7670.txt

=== SOURCES ===
S1: GEO10072.txt
S2: GEO19188.txt
S3: GEO18842.txt
S4: GEO19804.txt

=== Robust Rules ===
S1 1. ((EDNRB = -inf..7.502)) ==> (@Class = CASE)
   CF=0.994, PV=0E0, TP=173, FP=0, Pos=230, Neg=183
S2 2. ((AQP1 = -inf..10.879)) ==> (@Class = CASE)
   CF=0.993, PV=0E0, TP=147, FP=0, Pos=172, Neg=134
S3 3. ((GAPDH = 13.137..inf)) ==> (@Class = CASE)
   CF=0.986, PV=0E0, TP=71, FP=0, Pos=81, Neg=69
S3 4. ((CLDN18 = 10.750..inf)) ==> (@Class = CONTROL)
   CF=0.974, PV=0E0, TP=68, FP=1, Pos=69, Neg=81
S1 3. ((ARHGAP6 = -inf..6.246)) ==> (@Class = CASE)
   CF=0.987, PV=0E0, TP=177, FP=1, Pos=230, Neg=183
S3 4. ((CENPF = 6.523..inf)) ==> (@Class = CASE)
```

```
   CF=0.972, PV=0E0, TP=75, FP=1, Pos=81, Neg=69
S2 5. ((ADRB2 = -inf..8.585)) ==> (@Class = CASE)
   CF=0.969, PV=0E0, TP=150, FP=3, Pos=172, Neg=134
S2 6. ((AGER = 10.009..inf)) ==> (@Class = CONTROL)
   CF=0.963, PV=0E0, TP=125, FP=5, Pos=134, Neg=172
S2 4. ((FABP4 = -inf..9.073)) ==> (@Class = CASE)
   CF=0.968, PV=0E0, TP=147, FP=3, Pos=172, Neg=134
S1 5. ((CDH5 = -inf..8.460)) ==> (@Class = CASE)
   CF=0.966, PV=0E0, TP=208, FP=5, Pos=230, Neg=183
S2 7. ((CITED2 = -inf..8.506)) ==> (@Class = CASE)
   CF=0.959, PV=0E0, TP=112, FP=3, Pos=172, Neg=134
S2 8. ((CAV1 = 12.040..inf)) ==> (@Class = CONTROL)
   CF=0.958, PV=0E0, TP=112, FP=5, Pos=134, Neg=172
S2 3. ((ALOX5 = -inf..9.675)) ==> (@Class = CASE)
   CF=0.984, PV=0E0, TP=141, FP=1, Pos=172, Neg=134
S2 2. ((ABCA8 = -inf..7.460)) ==> (@Class = CASE)
   CF=0.993, PV=0E0, TP=144, FP=0, Pos=172, Neg=134
S2 3. ((CACYBP = 9.938..inf)) ==> (@Class = CASE)
   CF=0.967, PV=0E0, TP=102, FP=2, Pos=172, Neg=135
S3 2. ((COX7A1 = -inf..8.840)) ==> (@Class = CASE)
   CF=0.987, PV=0E0, TP=73, FP=0, Pos=81, Neg=70
S1 1. ((FRY = -inf..8.276)) ==> (@Class = CASE)
   CF=0.993, PV=0E0, TP=150, FP=0, Pos=230, Neg=184
S1 5. ((GPM6B = -inf..7.757)) ==> (@Class = CASE)
   CF=0.97, PV=0E0, TP=156, FP=3, Pos=231, Neg=183
S2 1. ((CELF2 = -inf..10.709)) ==> (@Class = CASE)
   CF=0.991, PV=0E0, TP=114, FP=0, Pos=173, Neg=134
```

F.2.5 MIXED CANCER

```
 === TARGET ===
GEO4412.txt

=== SOURCES ===
S1: GEO6956.txt
S2: GEO7670.txt
S3: GEO9348.txt
S4: GEO7904.txt

=== Robust Rules ===
S2 8. ((ABLIM1 = 14.070..inf)) ==> (@Class = CONTROL)
   CF=0.852, PV=2.45093E-11, TP=39, FP=15, Pos=81, Neg=205
S1 7. ((ATP5C1 = -inf..14.046)) ==> (@Class = CASE)
   CF=0.895, PV=1.58572E-7, TP=77, FP=3, Pos=274, Neg=101
S3 5. ((CEBPB = -inf..13.019)) ==> (@Class = CONTROL)
   CF=0.847, PV=3.15009E-9, TP=28, FP=13, Pos=54, Neg=166
S2 4. ((CCT6A = 12.695..inf)) ==> (@Class = CASE)
   CF=0.871, PV=2.25246E-8, TP=91, FP=5, Pos=205, Neg=81
S2 7. ((CALU = 13.234..inf)) ==> (@Class = CASE)
   CF=0.863, PV=3.5973E-10, TP=116, FP=7, Pos=205, Neg=82
S1 8. ((ALG3 = 11.139..inf) (CLIC1 = -inf..12.174)) ==> (@Class = CASE)
   CF=0.804, PV=7.60027E-5, TP=69, FP=6, Pos=274, Neg=102

=== TARGET ===
GEO7904.txt

=== SOURCES ===
S1: GEO6956.txt
```

```
S2: GEO4412.txt
S3: GEO7670.txt
S4: GEO9348.txt

=== Robust Rules ===
S2 2. ((ABCG2 = -inf..5.029)) ==> (@Class = CASE)
   CF=0.901, PV=2.59581E-12, TP=122, FP=5, Pos=206, Neg=82
S2 2. ((ADORA2B = -inf..7.933) (ARHGEF6 = -inf..8.822)) ==> (@Class = CASE)
   CF=0.863, PV=4.12895E-10, TP=116, FP=7, Pos=206, Neg=82
S2 3. ((ALPL = -inf..5.884) (ARHGEF6 = -inf..8.822)) ==> (@Class = CASE)
   CF=0.846, PV=2.35369E-9, TP=115, FP=8, Pos=206, Neg=82
S3 4. ((ACTG2 = -inf..9.285) (ANXA3 = -inf..7.756)) ==> (@Class = CASE)
   CF=0.862, PV=2.5524E-6, TP=71, FP=4, Pos=148, Neg=56

=== TARGET ===
GEO6956.txt

=== SOURCES ===
S1: GEO4412.txt
S2: GEO7904.txt
S3: GEO7670.txt
S4: GEO9348.txt

=== Robust Rules===
S3 1. ((CCT3 = 10.473..inf)) ==> (@Class = CASE)
   CF=0.949, PV=1.60871E-13, TP=129, FP=2, Pos=171, Neg=57
S3 3. ((CCL23 = 3.852..inf)) ==> (@Class = CONTROL)
   CF=0.884, PV=3.58558E-12, TP=32, FP=10, Pos=57, Neg=171
S2 3. ((GNG11 = 9.088..inf)) ==> (@Class = CONTROL)
   CF=0.847, PV=7.53639E-10, TP=32, FP=14, Pos=76, Neg=214
```

S3 4. ((FABP4 = -inf..4.882)) ==> (@Class = CASE)

CF=0.922, PV=2.84162E-11, TP=118, FP=3, Pos=171, Neg=57

F.2.6 PROSTATE CANCER

=== TARGET ===

GEO17951.txt

=== SOURCES ===

S1: GEO32448.txt

S2: GEO6956.txt

S3: GEO46602.txt

S4: GEO82188.txt

=== Robust Rules ===

S1 2. ((ERG = 8.766..inf)) ==> (@Class = CASE)

CF=0.966, PV=0E0, TP=100, FP=2, Pos=271, Neg=208

S2 13. ((GOLM1 = 11.838..inf)) ==> (@Class = CASE)

CF=0.914, PV=0E0, TP=112, FP=7, Pos=231, Neg=168

S3 15. ((GPRC5B = 8.518..inf)) ==> (@Class = CONTROL)

CF=0.891, PV=0E0, TP=89, FP=11, Pos=148, Neg=162

S2 14. ((ANXA2 = -inf..11.597)) ==> (@Class = CASE)

CF=0.953, PV=0E0, TP=104, FP=3, Pos=231, Neg=167

S1 2. ((B3GAT1 = 7.493..inf)) ==> (@Class = CASE)

CF=0.975, PV=0E0, TP=88, FP=1, Pos=271, Neg=207

S2 18. ((ESD = -inf..11.029)) ==> (@Class = CASE)

CF=0.827, PV=8.91154E-10, TP=83, FP=12, Pos=231

S3 15. ((GPRC5B = 8.486..inf)) ==> (@Class = CONTROL)

CF=0.87, PV=1.11022E-16, TP=90, FP=14, Pos=147

```
=== TARGET ===
GEO82188.txt

=== SOURCES ===
S1: GEO46602.txt
S2: GEO6956.txt
S3: GEO17951.txt
S4: GEO32448.txt

=== Robust Rules ===
S2 5. ((CSRP2 = -inf..8.918)) ==> (@Class = CASE)
   CF=0.938, PV=0E0, TP=108, FP=5, Pos=236, Neg=192
S3 10. ((DNASE2B = -inf..5.417)) ==> (@Class = CONTROL)
   CF=0.895, PV=5.21061E-12, TP=60, FP=6, Pos=172, Neg=167
S3 7. ((ABCC4 = 10.220..inf)) ==> (@Class = CASE)
   CF=0.913, PV=0E0, TP=100, FP=9, Pos=166, Neg=173
S3 13. ((DPYSL3 = 10.645..inf)) ==> (@Class = CONTROL)
   CF=0.847, PV=7.18314E-14, TP=91, FP=15, Pos=173, Neg=166
S2 14. ((ERBB3 = -inf..9.063)) ==> (@Class = CONTROL)
   CF=0.846, PV=4.10783E-15, TP=86, FP=18, Pos=193, Neg=235
S1 13. ((ARMCX1 = -inf..7.677)) ==> (@Class = CASE)
   CF=0.867, PV=2.22045E-16, TP=116, FP=13, Pos=271, Neg=207
S3 16. ((ACTA2 = 13.306..inf)) ==> (@Class = CONTROL)
   CF=0.852, PV=1.33227E-15, TP=101, FP=16, Pos=173, Neg=166
S2 7. ((ERG = 7.839..inf)) ==> (@Class = CASE)
   CF=0.898, PV=1.73195E-14, TP=83, FP=7, Pos=236, Neg=193
S2 8. ((CSRP2 = -inf..8.935)) ==> (@Class = CASE)
   CF=0.89, PV=0E0, TP=116, FP=11, Pos=236, Neg=193
S2 9. ((DUS1L = 9.893..inf)) ==> (@Class = CASE)
   CF=0.862, PV=3.0087E-14, TP=97, FP=12, Pos=236, Neg=193
```

S3 18. ((ALDH1A2 = -inf..8.107)) ==> (@Class = CASE)

CF=0.845, PV=7.01539E-12, TP=73, FP=13, Pos=167, Neg=173

BIBLIOGRAPHY

[1] Brazma A, Hingamp P, Quackenbush J, Sherlock G, Spellman P, Stoeckert C, et al. Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. Nature Genetics. 2001 Dec;29(4):365–371.

[2] Edgar R, Domrachev M, Lash AE. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. Nucleic acids research. 2002;30(1):207–210.

[3] Brazma A, Parkinson H, Sarkans U, Shojatalab M, Vilo J, Abeygunawardena N, et al. ArrayExpressa public repository for microarray gene expression data at the EBI. Nucleic acids research. 2003;31(1):68–71.

[4] Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, et al. Gene Ontology: tool for the unification of biology. Nature genetics. 2000;25(1):25–29.

[5] Kanehisa M, Goto S. KEGG: kyoto encyclopedia of genes and genomes. Nucleic acids research. 2000;28(1):27–30.

[6] National Research Council (US) Committee on A Framework for Developing a New Taxonomy of Disease. Toward Precision Medicine: Building a Knowledge Network for Biomedical Research and a New Taxonomy of Disease. The National Academies Collection: Reports funded by National Institutes of Health. Washington (DC): National Academies Press (US); 2011.

[7] Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, et al. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science. 1999 Oct;286(5439):531–537.

[8] van 't Veer LJ, Dai H, van de Vijver MJ, He YD, Hart AAM, Mao M, et al. Gene expression profiling predicts clinical outcome of breast cancer. Nature. 2002 Jan;415(6871):530–536.

[9] James G, Witten D, Hastie T, Tibshirani R. Statistical Learning. In: An Introduction to Statistical Learning. No. 103 in Springer Texts in Statistics. Springer New York; 2013. p. 15–57.

[10] Ein-Dor L, Zuk O, Domany E. Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. Proceedings of the National Academy of Sciences. 2006;103(15):5923–5928.

[11] Oberg AL, Bot BM, Grill DE, Poland GA, Therneau TM. Technical and biological variance structure in mRNA-Seq data: life in the real world. BMC genomics. 2012;13:304.

[12] Molloy MP, Brzezinski EE, Hang J, McDowell MT, VanBogelen RA. Overcoming technical variation and biological variation in quantitative proteomics. Proteomics. 2003 Oct;3(10):1912–1919.

[13] Lossos IS, Czerwinski DK, Alizadeh AA, Wechser MA, Tibshirani R, Botstein D, et al. Prediction of survival in diffuse large-B-cell lymphoma based on the expression of six genes. The New England Journal of Medicine. 2004 Apr;350(18):1828–1837.

[14] Fishel I, Kaufman A, Ruppin E. Meta-analysis of gene expression data: a predictor-based approach. Bioinformatics. 2007 Jul;23(13):1599–1606.

[15] McCloskey M, Cohen NJ. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. Psychology of Learning and Motivation. 1989 Dec;24:109–164.

[16] Larraaga P, Calvo B, Santana R, Bielza C, Galdiano J, Inza I, et al. Machine learning in bioinformatics. Briefings in Bioinformatics. 2006 Mar;7(1):86–112.

[17] Quinlan JR. C4. 5: programs for machine learning. vol. 1. Morgan kaufmann; 1993.

[18] Clearwater SH, Provost FJ. RL4: a tool for knowledge-based induction. ,Proceedings of the 2nd International IEEE Conference on Tools for Artificial Intelligence, 1990. 1990 Nov;p. 24–30.

[19] Frnkranz J, Gamberger D, Lavrac N. Foundations of Rule Learning. Springer Publishing Company, Incorporated; 2012.

[20] Ramasamy A, Mondry A, Holmes CC, Altman DG. Key Issues in Conducting a Meta-Analysis of Gene Expression Microarray Datasets. PLoS Medicine. 2008 Sep;5(9).

[21] Tseng GC, Ghosh D, Feingold E. Comprehensive literature review and statistical considerations for microarray meta-analysis. Nucleic acids research. 2012;p. gkr1265.

[22] Lazar C, Meganck S, Taminau J, Steenhoff D, Coletta A, Molter C, et al. Batch effect removal methods for microarray gene expression data integration: a survey. Briefings in Bioinformatics. 2013 Jul;14(4):469–490.

[23] Ganchev P, Malehorn D, Bigbee WL, Gopalakrishnan V. Transfer learning of classification rules for biomarker discovery and verification from molecular profiling studies. Journal of Biomedical Informatics. 2011 Dec;44, Supplement 1:S17–S23.

[24] Pan SJ, Yang Q. A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering. 2010 Oct;22(10):1345–1359.

[25] Ein-Dor L, Kela I, Getz G, Givol D, Domany E. Outcome signature genes in breast cancer: is there a unique set? Bioinformatics. 2005;21(2):171–178.

[26] Fan C, Oh DS, Wessels L, Weigelt B, Nuyten DS, Nobel AB, et al. Concordance among gene-expressionbased predictors for breast cancer. New England Journal of Medicine. 2006;355(6):560–569.

[27] Fridman JS, Lowe SW. Control of apoptosis by p53. Oncogene. 2003 Dec;22(56):9030–9040.

[28] Ganchev P. Transfer rule learning for biomarker discovery and verification from related data sets [University of Pittsburgh ETD]; 2011.

[29] Bigbee WL, Gopalakrishnan V, Weissfeld JL, Wilson DO, Dacic S, Lokshin AE, et al. 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. 2012;7(4):698.

[30] Gopalakrishnan V, Ganchev P, Ranganathan S, Bowser R. Rule learning for disease-specific biomarker discovery from clinical proteomic mass spectra. In: Data Mining for Biomedical Applications. Springer; 2006. p. 93–105.

[31] Ranganathan S, Williams E, Ganchev P, Gopalakrishnan V, Lacomis D, Urbinelli L, et al. Proteomic profiling of cerebrospinal fluid identifies biomarkers for amyotrophic lateral sclerosis. Journal of neurochemistry. 2005;95(5):1461–1471.

[32] Ryberg H, An J, Darko S, Lustgarten JL, Jaffa M, Gopalakrishnan V, et al. Discovery and verification of amyotrophic lateral sclerosis biomarkers by proteomics. Muscle & nerve. 2010;42(1):104–111.

[33] Caruana R. Multitask Learning. Machine Learning. 1997 Jul;28(1):41–75.

[34] Blitzer J. Domain adaptation of natural language processing systems. Dissertations available from ProQuest. 2008 Jan;p. 1–95.

[35] The Cancer Genome Atlas Research Network, Weinstein JN, Collisson EA, Mills GB, Shaw KRM, Ozenberger BA, et al. The Cancer Genome Atlas Pan-Cancer analysis project. Nature Genetics. 2013 Oct;45(10):1113–1120.

[36] Ciriello G, Miller ML, Aksoy BA, Senbabaoglu Y, Schultz N, Sander C. Emerging landscape of oncogenic signatures across human cancers. Nature Genetics. 2013 Oct;45(10):1127–1133.

[37] Schulze A, Downward J. Navigating gene expression using microarrays–a technology review. Nature Cell Biology. 2001 Aug;3(8):E190–195.

[38] Slonim DK, Yanai I. Getting started in gene expression microarray analysis. PLoS computational biology. 2009 Oct;5(10):e1000543.

[39] Walker MS, Hughes TA. Messenger RNA expression profiling using DNA microarray technology: diagnostic tool, scientific analysis or un-interpretable data? International Journal of Molecular Medicine. 2008 Jan;21(1):13–17.

[40] Lockhart DJ, Winzeler EA. Genomics, gene expression and DNA arrays. Nature. 2000 Jun;405(6788):827–836.

[41] Lander ES. Array of hope. Nature Genetics. 1999 Jan;21(1 Suppl):3–4.

[42] VanGuilder HD, Vrana KE, Freeman WM. Twenty-five years of quantitative PCR for gene expression analysis. BioTechniques. 2008 Apr;44(5):619–626.

[43] Bertone P, Stolc V, Royce TE, Rozowsky JS, Urban AE, Zhu X, et al. Global Identification of Human Transcribed Sequences with Genome Tiling Arrays. Science. 2004 Dec;306(5705):2242–2246.

[44] Shendure J, Ji H. Next-generation DNA sequencing. Nature Biotechnology. 2008 Oct;26(10):1135–1145.

[45] Wang Z, Gerstein M, Snyder M. RNA-Seq: a revolutionary tool for transcriptomics. Nature Reviews Genetics. 2009 Jan;10(1):57–63.

[46] Ozsolak F, Milos PM. RNA sequencing: advances, challenges and opportunities. Nature Reviews Genetics. 2011 Feb;12(2):87–98.

[47] Zhao S, Fung-Leung WP, Bittner A, Ngo K, Liu X. Comparison of RNA-Seq and microarray in transcriptome profiling of activated T cells. PloS One. 2014;9(1):e78644.

[48] Xu X, Zhang Y, Williams J, Antoniou E, McCombie WR, Wu S, et al. Parallel comparison of Illumina RNA-Seq and Affymetrix microarray platforms on transcriptomic profiles generated from 5-aza-deoxy-cytidine treated HT-29 colon cancer cells and simulated datasets. BMC Bioinformatics. 2013 Jun;14(Suppl 9):S1.

[49] Malone JH, Oliver B. Microarrays, deep sequencing and the true measure of the transcriptome. BMC Biology. 2011 May;9(1):34.

[50] Marioni JC, Mason CE, Mane SM, Stephens M, Gilad Y. RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. Genome Research. 2008 Sep;18(9):1509–1517.

[51] Seqc/Maqc-Iii Consortium. A comprehensive assessment of RNA-seq accuracy, reproducibility and information content by the Sequencing Quality Control Consortium. Nature Biotechnology. 2014 Sep;32(9):903–914.

[52] Rung J, Brazma A. Reuse of public genome-wide gene expression data. Nature Reviews Genetics. 2013 Feb;14(2):89–99.

[53] Churchill GA. Fundamentals of experimental design for cDNA microarrays. Nature Genetics. 2002 Dec;32 Suppl:490–495.

[54] Chen DT. A graphical approach for quality control of oligonucleotide array data. Journal of Biopharmaceutical Statistics. 2004 Aug;14(3):591–606.

[55] MAQC Consortium, Shi L, Reid LH, Jones WD, Shippy R, Warrington JA, et al. The MicroArray Quality Control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements. Nature Biotechnology. 2006 Sep;24(9):1151–1161.

[56] Quackenbush J. Microarray data normalization and transformation. Nature Genetics. 2002 Dec;32 Suppl:496–501.

[57] Yang YH, Dudoit S, Luu P, Lin DM, Peng V, Ngai J, et al. Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. Nucleic Acids Research. 2002 Feb;30(4):e15.

[58] Smyth GK, Speed T. Normalization of cDNA microarray data. Methods (San Diego, Calif). 2003 Dec;31(4):265–273.

[59] Qin LX, Kerr KF. Empirical evaluation of data transformations and ranking statistics for microarray analysis. Nucleic Acids Research. 2004;32(18):5471–5479.

[60] Rocke DM, Durbin B. Approximate variance-stabilizing transformations for gene-expression microarray data. Bioinformatics. 2003 May;19(8):966–972.

[61] Pounds S, Cheng C. Statistical development and evaluation of microarray gene expression data filters. Journal of Computational Biology: A Journal of Computational Molecular Cell Biology. 2005 May;12(4):482–495.

[62] Gentleman R. Design and Analysis of DNA Microarray Investigations. Richard M. Simon, Edward L. Korn, Lisa M. McShane, Michael D. Radmacher, George W. Wright, and Yingdong Zhao. Journal of the American Statistical Association. 2005;100(June):711–712.

[63] Pan W. A comparative review of statistical methods for discovering differentially expressed genes in replicated microarray experiments. Bioinformatics (Oxford, England). 2002 Apr;18(4):546–554.

[64] Cui X, Hwang JTG, Qiu J, Blades NJ, Churchill GA. Improved statistical tests for differential gene expression by shrinking variance components estimates. Biostatistics (Oxford, England). 2005 Jan;6(1):59–75.

[65] Barrell D, Dimmer E, Huntley RP, Binns D, ODonovan C, Apweiler R. The GOA database in 2009an integrated Gene Ontology Annotation resource. Nucleic acids research. 2009;37(suppl 1):D396–D403.

[66] Tamayo P, Steinhardt G, Liberzon A, Mesirov JP. The Limitations of Simple Gene Set Enrichment Analysis Assuming Gene Independence. ArXiv e-prints. 2011 Oct;1110:4128.

[67] Tarca AL, Bhatti G, Romero R. A Comparison of Gene Set Analysis Methods in Terms of Sensitivity, Prioritization and Specificity. PLoS ONE. 2013 Nov;8(11):e79217.

[68] Hung JH, Yang TH, Hu Z, Weng Z, DeLisi C. Gene set enrichment analysis: performance evaluation and usage guidelines. Briefings in Bioinformatics. 2012 May;13(3):281–291.

[69] Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, et al. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences. 2005 Oct;102(43):15545–15550.

[70] Barabasi AL, Oltvai ZN. Network biology: understanding the cell's functional organization. Nature Reviews Genetics. 2004;5(2):101–113.

[71] Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U. Network motifs: simple building blocks of complex networks. Science (New York, NY). 2002 Oct;298(5594):824–827.

[72] Ravasz E, Somera AL, Mongru DA, Oltvai ZN, Barabsi AL. Hierarchical Organization of Modularity in Metabolic Networks. Science. 2002 Aug;297(5586):1551–1555.

[73] Ma X, Gao L. Biological network analysis: insights into structure and functions. Briefings in Functional Genomics. 2012 Nov;11(6):434–442.

[74] Alizadeh AA, Eisen MB, Davis RE, Ma C, Lossos IS, Rosenwald A, et al. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature. 2000 Feb;403(6769):503–511.

[75] Milligan GW, Cooper MC. An examination of procedures for determining the number of clusters in a data set. Psychometrika. 1985 Jun;50(2):159–179.

[76] Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics. 1987;20:53–65.

[77] Kerr MK, Churchill GA. Bootstrapping cluster analysis: Assessing the reliability of conclusions from microarray experiments. Proceedings of the National Academy of Sciences. 2001 Jul;98(16):8961–8965.

[78] Tibshirani R, Walther G, Hastie T. Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2001 Jan;63(2):411–423.

[79] Garge NR, Page GP, Sprague AP, Gorman BS, Allison DB. Reproducible Clusters from Microarray Research: Whither? BMC Bioinformatics. 2005 Jul;6(Suppl 2):S10.

[80] Dudoit S, Fridlyand J, Speed TP. Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data. Journal of the American Statistical Association. 2002 Mar;97(457):77–87.

[81] Lee JW, Lee JB, Park M, Song SH. An extensive comparison of recent classification tools applied to microarray data. Computational Statistics & Data Analysis. 2005 Apr;48(4):869–885.

[82] Kittler J. Combining classifiers: A theoretical framework. Pattern Analysis and Applications. 1998 Mar;1(1):18–27.

[83] Kittler J. A Framework for Classifier Fusion: Is It Still Needed? In: Ferri FJ, Iesta JM, Amin A, Pudil P, editors. Advances in Pattern Recognition. No. 1876 in Lecture Notes in Computer Science. Springer Berlin Heidelberg; 2000. p. 45–56.

[84] Daz-Uriarte R. Supervised Methods with Genomic Data: a Review and Cautionary View. In: Azuaje F, Dopazo J, editors. Data Analysis and Visualization in Genomics and Proteomics. John Wiley & Sons, Ltd; 2005. p. 193–214.

[85] Kumar Sarmah C, Samarasinghe S. Microarray Data Integration: Frameworks and a List of Underlying Issues. Current Bioinformatics. 2010 Dec;5(4):280–289.

[86] Li J, Tseng GC. An adaptively weighted statistic for detecting differential gene expression when combining multiple transcriptomic studies. The Annals of Applied Statistics. 2011 Jun;5(2A):994–1019. ArXiv: 1108.3180.

[87] Choi JK, Yu U, Kim S, Yoo OJ. Combining multiple microarray studies and modeling interstudy variation. Bioinformatics (Oxford, England). 2003;19 Suppl 1:i84–90.

[88] Hong F, Breitling R, McEntee CW, Wittner BS, Nemhauser JL, Chory J. RankProd: a bioconductor package for detecting differentially expressed genes in meta-analysis. Bioinformatics (Oxford, England). 2006 Nov;22(22):2825–2827.

[89] Song C, Tseng GC. Hypothesis setting and order statistic for robust genomic meta-analysis. The Annals of Applied Statistics. 2014 Jun;8(2):777–800. ArXiv: 1407.8376.

[90] Hwang KB, Kong SW, Greenberg SA, Park PJ. Combining gene expression data from different generations of oligonucleotide arrays. BMC bioinformatics. 2004 Oct;5:159.

[91] Cheadle C, Vawter MP, Freed WJ, Becker KG. Analysis of microarray data using Z score transformation. The Journal of molecular diagnostics: JMD. 2003 May;5(2):73–81.

[92] Marron JS, Todd MJ, Ahn J. Distance-Weighted Discrimination. Journal of the American Statistical Association. 2007 Dec;102(480):1267–1271.

[93] Benito M, Parker J, Du Q, Wu J, Xiang D, Perou CM, et al. Adjustment of systematic microarray data biases. Bioinformatics. 2004 Jan;20(1):105–114.

[94] Johnson WE, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical Bayes methods. Biostatistics. 2007 Jan;8(1):118–127.

[95] Jiang H, Deng Y, Chen HS, Tao L, Sha Q, Chen J, et al. Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. BMC Bioinformatics. 2004 Jun;5(1):81.

[96] Shabalin AA, Tjelmeland H, Fan C, Perou CM, Nobel AB. Merging two gene-expression studies via cross-platform normalization. Bioinformatics. 2008 May;24(9):1154–1160.

[97] Xiong H, Zhang Y, Chen XW, Yu J. Cross-platform microarray data integration using the normalised linear transform. International Journal of Data Mining and Bioinformatics. 2010;4(2):142–157.

[98] Rudy J, Valafar F. Empirical comparison of cross-platform normalization methods for gene expression data. BMC Bioinformatics. 2011 Dec;12(1):467.

[99] Polikar R. Ensemble based systems in decision making. IEEE Circuits and Systems Magazine. 2006;6(3):21–45.

[100] Breiman L, Breiman L. Bagging Predictors. In: Machine Learning; 1996. p. 123–140.

[101] Freund Y, Schapire RE. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. J Comput Syst Sci. 1997 Aug;55(1):119–139.

[102] Jacobs RA, Jordan MI, Nowlan SJ, Hinton GE. Adaptive Mixtures of Local Experts. Neural Comput. 1991 Mar;3(1):79–87.

[103] Buxton BF, Langdon WB, Barrett SJ. Data Fusion by Intelligent Classifier Combination. Measurement and Control. 2001 Oct;34(8):229–234.

[104] Azuaje F. Bioinformatics and biomarker discovery:" omic" data analysis for personalized medicine. John Wiley & Sons; 2011.

[105] Hamid JS, Greenwood CMT, Beyene J. Weighted kernel Fisher discriminant analysis for integrating heterogeneous data. Computational Statistics & Data Analysis. 2012 Jun;56(6):2031–2040.

[106] Wang X, Xing EP, Schaid DJ. Kernel methods for large-scale genomic data analysis. Briefings in Bioinformatics. 2015 Mar;16(2):183–192.

[107] Mehta S, Shelling A, Muthukaruppan A, Lasham A, Blenkiron C, Laking G, et al. Predictive and prognostic molecular markers for cancer medicine. Therapeutic Advances in Medical Oncology. 2010 Mar;2(2):125–148.

[108] Warnat P, Eils R, Brors B. Cross-platform analysis of cancer microarray data improves gene expression based classification of phenotypes. BMC Bioinformatics. 2005 Nov;6(1):265.

[109] Zhang Z, Chen D, Fenstermacher DA. Integrated analysis of independent gene expression microarray datasets improves the predictability of breast cancer outcome. BMC Genomics. 2007 Sep;8:331.

[110] Daemen A, Gevaert O, Ojeda F, Debucquoy A, Suykens JA, Sempoux C, et al. A kernel-based integration of genome-wide data for clinical decision support. Genome Medicine. 2009 Apr;1(4):39.

[111] Lanckriet GRG, De Bie T, Cristianini N, Jordan MI, Noble WS. A statistical framework for genomic data fusion. Bioinformatics (Oxford, England). 2004 Nov;20(16):2626–2635.

[112] Ptitsyn AA, Weil MM, Thamm DH. Systems biology approach to identification of biomarkers for metastatic progression in cancer. BMC bioinformatics. 2008;9 Suppl 9:S8.

[113] Knickerbocker T, Chen JR, Thadhani R, MacBeath G. An integrated approach to prognosis using protein microarrays and nonparametric methods. Molecular Systems Biology. 2007 Jun;3.

[114] Taylor ME, Stone P. Transfer Learning for Reinforcement Learning Domains: A Survey. J Mach Learn Res. 2009 Dec;10:1633–1685.

[115] Rosenstein MT, Marx Z, Kaelbling LP, Dietterich TG. To transfer or not to transfer. In: In NIPS05 Workshop, Inductive Transfer: 10 Years Later; 2005. .

[116] Daume III H, Marcu D. Domain Adaptation for Statistical Classifiers. arXiv:11096341 [cs]. 2011 Sep;ArXiv: 1109.6341.

[117] Pan SJ, Kwok JT, Yang Q. Transfer Learning via Dimensionality Reduction. In: Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2. AAAI'08. Chicago, Illinois: AAAI Press; 2008. p. 677–682.

[118] Wang Z, Song Y, Zhang C. Transferred Dimensionality Reduction. In: Daelemans W, Goethals B, Morik K, editors. Machine Learning and Knowledge Discovery in Databases. No. 5212 in Lecture Notes in Computer Science. Springer Berlin Heidelberg; 2008. p. 550–565.

[119] Torrey L, Shavlik J, Natarajan S, Kuppili P, Walker T. Transfer in reinforcement learning via Markov logic networks. In: AAAI Workshop on Transfer Learning for Complex Tasks; 2008. .

[120] Dai W, Yang Q, Xue Gr, Yu Y. Boosting for transfer learning. In: In ICML; 2007. .

[121] Jiang J, Zhai C. Instance weighting for domain adaptation in NLP. In: In ACL 2007; 2007. p. 264–271.

[122] Blitzer J, Dredze M, Pereira F. Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In: In ACL; 2007. p. 187–205.

[123] Argyriou A, Pontil M, Micchelli CA, Ying Y. A spectral regularization framework for multi-task structure learning. 2009;.

[124] Lawrence ND, Platt JC. Learning to Learn with the Informative Vector Machine. In: In Proceedings of the International Conference in Machine Learning. Morgan Kaufmann; 2004. .

[125] Gao J, Fan W, Jiang J, Han J. Knowledge transfer via multiple model local structure mapping. In: In International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV; 2008. .

[126] Mihalkova L, Huynh T, Mooney RJ. Mapping and revising markov logic networks for transfer learning. In: In Proceedings of the 22 nd National Conference on Artificial Intelligence (AAAI; 2007. p. 608–614.

[127] Davis J, Domingos P. Deep transfer via second-order markov logic. In: In Proceedings of the AAAI Workshop on Transfer Learning For Complex Tasks; 2008. .

[128] Jebara T. Multi-Task Feature and Kernel Selection for SVMs. In: Proc. of ICML 2004; 2004. .

[129] Xu Q, Yang Q. A Survey of Transfer and Multitask Learning in Bioinformatics. Journal of Computing Science and Engineering. 2011 Aug;.

[130] Widmer C, Rtsch G. Multitask Learning in Computational Biology. In: ICML Unsupervised and Transfer Learning; 2012. p. 207–216.

[131] Buchanan BG, Feigenbaum EA. Dendral and meta-dendral: Their applications dimension. Artificial Intelligence. 1978 Aug;11(12):5–24.

[132] Provost F, Aronis JM, Buchanan BG. Rule-Space Search for Knowledge-Based Discovery. In: CIIO Working Paper IS 99-012, Stern School of Business; 1999. p. 1001–2.

[133] Clark P, Niblett T. The CN2 Induction Algorithm. In: Machine Learning; 1989. p. 261–283.

[134] Vanathi Gopalakrishnan PG. Rule Learning for Disease-Specific Biomarker Discovery from Clinical Proteomic Mass Spectra. 2006;p. 93–105.

[135] Ogoe HA, Visweswaran S, Lu X, Gopalakrishnan V. Knowledge transfer via classification rules using functional mapping for integrative modeling of gene expression data. BMC bioinformatics. 2015;16:226.

[136] Liu1 G, Kong L, Gopalakrishnan V. A Partitioning Based Adaptive Method for Robust Removal of Irrelevant Features from High-dimensional Biomedical Datasets. AMIA Summits on Translational Science Proceedings. 2012 Mar;2012:52–61.

[137] Saeys Y, Inza I, Larraaga P. A review of feature selection techniques in. Bioinformatics. 2007 Oct;23(19):2507–2517.

[138] Yang Y, Webb GI. On Why Discretization Works for Naive-Bayes Classifiers. In: Gedeon TTD, Fung LCC, editors. AI 2003: Advances in Artificial Intelligence. No. 2903 in Lecture Notes in Computer Science. Springer Berlin Heidelberg; 2003. p. 440–452.

[139] Garcia S, Luengo J, Sez JA, Lpez V, Herrera F. A Survey of Discretization Techniques: Taxonomy and Empirical Analysis in Supervised Learning. IEEE Transactions on Knowledge and Data Engineering. 2013 Apr;25(4):734–750.

[140] Lustgarten JL, Gopalakrishnan V, Grover H, Visweswaran S. Improving classification performance with discretization on biomedical datasets. AMIA Annual Symposium proceedings / AMIA Symposium AMIA Symposium. 2008;p. 445–449.

[141] Lustgarten JL, Visweswaran S, Gopalakrishnan V, Cooper GF. Application of an efficient Bayesian discretization method to biomedical data. BMC bioinformatics. 2011;12(1):309.

[142] Fayyad UM, Irani KB. Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In: Bajcsy R, editor. Proceedings of the 13th International Joint Conference on Artificial Intelligence. Chambry, France, August 28 - September 3, 1993. Morgan Kaufmann; 1993. p. 1022–1029.

[143] Wong AKC, Chiu DKY. Synthesizing Statistical Knowledge from Incomplete Mixed-Mode Data. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1987 Nov;PAMI-9(6):796–805.

[144] Monti S, Cooper GF. A multivariate discretization method for learning Bayesian networks from mixed data. In: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc.; 1998. p. 404–413.

[145] Provost F, Aronis JM, Buchanan BG. Rule-Space Search for Knowledge-Based Discovery. In: CIIO Working Paper IS 99-012, Stern School of Business; 1999. p. 1001–2.

[146] Fawcett T. Using rule sets to maximize ROC performance. In: ICDM 2001, Proceedings IEEE International Conference on Data Mining, 2001; 2001. p. 131–138.

[147] Ratcliff R. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. Psychological Review. 1990 Apr;97(2):285–308.

[148] Barrell D, Dimmer E, Huntley RP, Binns D, O'Donovan C, Apweiler R. The GOA database in 2009–an integrated Gene Ontology Annotation resource. Nucleic Acids Research. 2009 Jan;37(Database issue):D396–403.

[149] Wang JZ, Du Z, Payattakool R, Yu PS, Chen CF. A new method to measure the semantic similarity of GO terms. Bioinformatics. 2007 May;23(10):1274–1281.

[150] Ng AY, Jordan MI, Weiss Y, others. On spectral clustering: Analysis and an algorithm. Advances in neural information processing systems. 2002;2:849–856.

[151] Hanahan D, Weinberg RA. Hallmarks of cancer: the next generation. Cell. 2011;144(5):646–674.

[152] Ooi CH, Ivanova T, Wu J, Lee M, Tan IB, Tao J, et al. Oncogenic Pathway Combinations Predict Clinical Prognosis in Gastric Cancer. PLoS Genet. 2009 Oct;5(10):e1000676.

[153] Huang S, Yee C, Ching T, Yu H, Garmire LX. A novel model to combine clinical and pathway-based transcriptomic information for the prognosis prediction of breast cancer. PLoS computational biology. 2014 Sep;10(9):e1003851.

[154] Shapiro P, Matheus C. The interestingness of deviations; 1994. .

[155] Geng L, Hamilton HJ. Interestingness Measures for Data Mining: A Survey. ACM Comput Surv. 2006 Sep;38(3).

[156] Pavlopoulou A, Spandidos DA, Michalopoulos I. Human cancer databases (review). Oncology Reports. 2015 Jan;33(1):3–18.

[157] Hanahan D, Weinberg RA. The hallmarks of cancer. Cell. 2000 Jan;100(1):57–70.

[158] Freije WA, Castro-Vargas FE, Fang Z, Horvath S, Cloughesy T, Liau LM, et al. Gene expression profiling of gliomas strongly predicts survival. Cancer Research. 2004 Sep;64(18):6503–6510.

[159] Phillips HS, Kharbanda S, Chen R, Forrest WF, Soriano RH, Wu TD, et al. Molecular subclasses of high-grade glioma predict prognosis, delineate a pattern of disease progression, and resemble stages in neurogenesis. Cancer Cell. 2006 Mar;9(3):157–173.

[160] Sun L, Hui AM, Su Q, Vortmeyer A, Kotliarov Y, Pastorino S, et al. Neuronal and glioma-derived stem cell factor induces angiogenesis within the brain. Cancer Cell. 2006 Apr;9(4):287–300.

[161] Petalidis LP, Oulas A, Backlund M, Wayland MT, Liu L, Plant K, et al. Improved grading and survival prediction of human astrocytic brain tumors by artificial neural network analysis of gene expression microarray data. Molecular Cancer Therapeutics. 2008 May;7(5):1013–1024.

[162] Gravendeel LAM, Kouwenhoven MCM, Gevaert O, de Rooi JJ, Stubbs AP, Duijm JE, et al. Intrinsic gene expression profiles of gliomas are a better predictor of survival than histology. Cancer Research. 2009 Dec;69(23):9065–9072.

[163] Pau Ni IB, Zakaria Z, Muhammad R, Abdullah N, Ibrahim N, Aina Emran N, et al. Gene expression patterns distinguish breast carcinomas from normal breast tissues: the Malaysian context. Pathology, Research and Practice. 2010 Apr;206(4):223–228.

[164] Clarke C, Madden SF, Doolan P, Aherne ST, Joyce H, O'Driscoll L, et al. Correlating transcriptional networks to breast cancer survival: a large-scale coexpression analysis. Carcinogenesis. 2013 Oct;34(10):2300–2308.

[165] Lopez FJ, Cuadros M, Cano C, Concha A, Blanco A. Biomedical application of fuzzy association rules for identifying breast cancer biomarkers. Medical & Biological Engineering & Computing. 2012 Sep;50(9):981–990.

[166] Richardson AL, Wang ZC, De Nicolo A, Lu X, Brown M, Miron A, et al. X chromosomal abnormalities in basal-like human breast cancer. Cancer Cell. 2006 Feb;9(2):121–132.

[167] Alhopuro P, Sammalkorpi H, Niittymki I, Bistrm M, Raitila A, Saharinen J, et al. Candidate driver genes in microsatellite-unstable colorectal cancer. International Journal of Cancer Journal International Du Cancer. 2012 Apr;130(7):1558–1566.

[168] Uddin S, Ahmed M, Hussain A, Abubaker J, Al-Sanea N, AbdulJabbar A, et al. Genome-wide expression analysis of Middle Eastern colorectal cancer reveals FOXM1 as a novel target for cancer therapy. The American Journal of Pathology. 2011 Feb;178(2):537–547.

[169] Skrzypczak M, Goryca K, Rubel T, Paziewska A, Mikula M, Jarosz D, et al. Modeling oncogenic signaling in colon tumors by multidirectional analyses of microarray data directed for maximization of analytical reliability. PloS One. 2010;5(10).

[170] Galamb O, Sipos F, Solymosi N, Spisk S, Krencs T, Tth K, et al. Diagnostic mRNA expression patterns of inflamed, benign, and malignant colorectal biopsy specimen and their correlation with peripheral blood results. Cancer Epidemiology, Biomarkers & Prevention: A Publication of the American Association for Cancer Research, Cosponsored by the American Society of Preventive Oncology. 2008 Oct;17(10):2835–2845.

[171] Hong Y, Downey T, Eu KW, Koh PK, Cheah PY. A 'metastasis-prone' signature for early-stage mismatch-repair proficient sporadic colorectal cancer patients and its implications for possible therapeutics. Clinical & Experimental Metastasis. 2010 Feb;27(2):83–90.

[172] Landi MT, Dracheva T, Rotunno M, Figueroa JD, Liu H, Dasgupta A, et al. Gene expression signature of cigarette smoking and its role in lung adenocarcinoma development and survival. PloS One. 2008;3(2):e1651.

[173] Sanchez-Palencia A, Gomez-Morales M, Gomez-Capilla JA, Pedraza V, Boyero L, Rosell R, et al. Gene expression profiling reveals novel biomarkers in nonsmall cell lung cancer. International Journal of Cancer Journal International Du Cancer. 2011 Jul;129(2):355–364.

[174] Hou J, Aerts J, den Hamer B, van Ijcken W, den Bakker M, Riegman P, et al. Gene expression-based classification of non-small cell lung carcinomas and survival prediction. PloS One. 2010;5(4):e10312.

[175] Lu TP, Tsai MH, Lee JM, Hsu CP, Chen PC, Lin CW, et al. Identification of a novel biomarker, SEMA5A, for non-small cell lung carcinoma in nonsmoking women. Cancer Epidemiology, Biomarkers & Prevention: A Publication of the American Association for Cancer Research, Cosponsored by the American Society of Preventive Oncology. 2010 Oct;19(10):2590–2597.

[176] Wallace TA, Prueitt RL, Yi M, Howe TM, Gillespie JW, Yfantis HG, et al. Tumor immunobiological differences in prostate cancer between African-American and European-American men. Cancer Research. 2008 Feb;68(3):927–936.

[177] Jia Z, Wang Y, Sawyers A, Yao H, Rahmatpanah F, Xia XQ, et al. Diagnosis of prostate cancer using differentially expressed genes in stroma. Cancer Research. 2011 Apr;71(7):2476–2487.

[178] Derosa CA, Furusato B, Shaheduzzaman S, Srikantan V, Wang Z, Chen Y, et al. Elevated osteonectin/SPARC expression in primary prostate cancer predicts metastatic progression. Prostate Cancer and Prostatic Diseases. 2012 Jun;15(2):150–156.

[179] Wang Y, Xia XQ, Jia Z, Sawyers A, Yao H, Wang-Rodriquez J, et al. In silico estimates of tissue components in surgical samples based on expression profiling data. Cancer Research. 2010 Aug;70(16):6448–6455.

[180] Cover TM, Thomas JA. Elements of Information Theory. Wiley; 2006.

[181] Kraskov A, Stgbauer H, Grassberger P. Estimating mutual information. Physical Review E. 2004 Jun;69(6):066138.

[182] Ross BC. Mutual Information between Discrete and Continuous Data Sets. PLOS ONE. 2014 Feb;9(2):e87357.

[183] Dionisio A, Menezes R, Mendes DA. Mutual information: a measure of dependency for nonlinear time series. Physica A: Statistical Mechanics and its Applications. 2004 Dec;344(12):326–329.

[184] Senner V, Ratzinger S, Mertsch S, Grssel S, Paulus W. Collagen XVI expression is upregulated in glioblastomas and promotes tumor cell adhesion. FEBS letters. 2008 Oct;582(23-24):3293–3300.

[185] Bauer R, Ratzinger S, Wales L, Bosserhoff A, Senner V, Grifka J, et al. Inhibition of collagen XVI expression reduces glioma cell invasiveness. Cellular Physiology and Biochemistry: International Journal of Experimental Cellular Physiology, Biochemistry, and Pharmacology. 2011;27(3-4):217–226.

[186] Logsdon CD, Fuentes MK, Huang EH, Arumugam T. RAGE and RAGE ligands in cancer. Current Molecular Medicine. 2007 Dec;7(8):777–789.

[187] Hudson BI, Carter AM, Harja E, Kalea AZ, Arriero M, Yang H, et al. Identification, classification, and expression of RAGE gene splice variants. FASEB journal: official publication of the Federation of American Societies for Experimental Biology. 2008 May;22(5):1572–1580.

[188] Malik P, Chaudhry N, Mittal R, Mukherjee TK. Role of receptor for advanced glycation end products in the complication and progression of various types of cancers. Biochimica Et Biophysica Acta. 2015 Sep;1850(9):1898–1904.

[189] Sparvero LJ, Asafu-Adjei D, Kang R, Tang D, Amin N, Im J, et al. RAGE (Receptor for Advanced Glycation Endproducts), RAGE ligands, and their role in cancer and inflammation. Journal of Translational Medicine. 2009;7:17.

[190] Englert JM, Hanford LE, Kaminski N, Tobolewski JM, Tan RJ, Fattman CL, et al. A role for the receptor for advanced glycation end products in idiopathic pulmonary fibrosis. The American Journal of Pathology. 2008 Mar;172(3):583–591.

[191] Bartling B, Hofmann HS, Weigle B, Silber RE, Simm A. Down-regulation of the receptor for advanced glycation end-products (RAGE) supports non-small cell lung carcinoma. Carcinogenesis. 2005 Feb;26(2):293–301.

[192] Klezovitch O, Chevillet J, Mirosevich J, Roberts RL, Matusik RJ, Vasioukhin V. Hepsin promotes prostate cancer progression and metastasis. Cancer Cell. 2004 Aug;6(2):185–195.

[193] Wu Q, Parry G. Hepsin and prostate cancer. Frontiers in Bioscience: A Journal and Virtual Library. 2007;12:5052–5059.

[194] Valkenburg KC, Hostetter G, Williams BO. Concurrent Hepsin overexpression and adenomatous polyposis coli deletion causes invasive prostate carcinoma in mice. The Prostate. 2015 Oct;75(14):1579–1585.

[195] Le PU, Angers-Loustau A, de Oliveira RMW, Ajlan A, Brassard CL, Dudley A, et al. DRR drives brain cancer invasion by regulating cytoskeletal-focal adhesion dynamics. Oncogene. 2010 Aug;29(33):4636–4647.

[196] Liu Q, Zhao XY, Bai RZ, Liang SF, Nie CL, Yuan Z, et al. Induction of tumor inhibition and apoptosis by a candidate tumor suppressor gene DRR1 on 3p21.1. Oncology Reports. 2009 Nov;22(5):1069–1075.

[197] Wang L, Darling J, Zhang JS, Liu W, Qian J, Bostwick D, et al. Loss of expression of the DRR 1 gene at chromosomal segment 3p21.1 in renal cell carcinoma. Genes, Chromosomes & Cancer. 2000 Jan;27(1):1–10.

[198] Stobdan T, Zhou D, Ao-Ieong E, Ortiz D, Ronen R, Hartley I, et al. Endothelin receptor B, a candidate gene from human studies at high altitude, improves cardiac tolerance to hypoxia in genetically engineered heterozygote mice. Proceedings of the National Academy of Sciences of the United States of America. 2015 Aug;112(33):10425–10430.

[199] Granstrm AL, Markljung E, Fink K, Nordenskjld E, Nilsson D, Wester T, et al. A novel stop mutation in the EDNRB gene in a family with Hirschsprung's disease associated with multiple sclerosis. Journal of Pediatric Surgery. 2014 Apr;49(4):622–625.

[200] Bessho A, Tabata M, Kiura K, Takata I, Nagata T, Fujimoto N, et al. Detection of occult tumor cells in peripheral blood from patients with small cell lung cancer by reverse transcriptase-polymerase chain reaction. Anticancer Research. 2000 Apr;20(2B):1149–1154.

[201] Nishimura S, Tsuda H, Ito K, Takano M, Terai Y, Jobo T, et al. Differential expression of hypoxia-inducible protein 2 among different histological types of epithelial ovarian cancer and in clear cell adenocarcinomas. International Journal of Gynecological Cancer: Official Journal of the International Gynecological Cancer Society. 2010 Feb;20(2):220–226.

[202] Nishimura S, Tsuda H, Nomura H, Kataoka F, Chiyoda T, Tanaka H, et al. Expression of hypoxia-inducible 2 (HIG2) protein in uterine cancer. European Journal of Gynaecological Oncology. 2011;32(2):146–149.

[203] Kim SH, Wang D, Park YY, Katoh H, Margalit O, Sheffer M, et al. HIG2 promotes colorectal cancer progression via hypoxia-dependent and independent pathways. Cancer Letters. 2013 Dec;341(2):159–165.

[204] Supuran CT. Carbonic anhydrases: novel therapeutic applications for inhibitors and activators. Nature Reviews Drug Discovery. 2008 Feb;7(2):168–181.

[205] Bootorabi F, Jnis J, Smith E, Waheed A, Kukkurainen S, Hytnen V, et al. Analysis of a shortened form of human carbonic anhydrase VII expressed in vitro compared to the full-length enzyme. Biochimie. 2010 Aug;92(8):1072–1080.

[206] Chu CM, Yao CT, Chang YT, Chou HL, Chou YC, Chen KH, et al. Gene expression profiling of colorectal tumors and normal mucosa by microarrays meta-analysis using prediction analysis of microarray, artificial neural network, classification, and regression trees. Disease Markers. 2014;2014:634123.

[207] Birkenkamp-Demtroder K, Olesen SH, Srensen FB, Laurberg S, Laiho P, Aaltonen LA, et al. Differential gene expression in colon cancer of the caecum versus the sigmoid and rectosigmoid. Gut. 2005 Mar;54(3):374–384.

[208] Yang GZ, Hu L, Cai J, Chen HY, Zhang Y, Feng D, et al. Prognostic value of carbonic anhydrase VII expression in colorectal carcinoma. BMC cancer. 2015;15:209.

[209] Larue L, Antos C, Butz S, Huber O, Delmas V, Dominis M, et al. A role for cadherins in tissue formation. Development (Cambridge, England). 1996 Oct;122(10):3185–3194.

[210] Imai K, Hirata S, Irie A, Senju S, Ikuta Y, Yokomine K, et al. Identification of a novel tumor-associated antigen, cadherin 3/P-cadherin, as a possible target for immunotherapy of pancreatic, gastric, and colorectal cancers. Clinical Cancer Research: An Official Journal of the American Association for Cancer Research. 2008 Oct;14(20):6487–6495.

[211] Broussard EK, Kim R, Wiley JC, Marquez JP, Annis JE, Pritchard D, et al. Identification of putative immunologic targets for colon cancer prevention based on conserved gene upregulation from preinvasive to malignant lesions. Cancer Prevention Research (Philadelphia, Pa). 2013 Jul;6(7):666–674.

[212] McCall MN, Uppal K, Jaffee HA, Zilliox MJ, Irizarry RA. The Gene Expression Barcode: leveraging public data repositories to begin cataloging the human and murine transcriptomes. Nucleic Acids Research. 2011 Jan;39(Database issue):D1011–1015.

[213] Barbero S, Mielgo A, Torres V, Teitz T, Shields DJ, Mikolon D, et al. Caspase-8 association with the focal adhesion complex promotes tumor cell migration and metastasis. Cancer Research. 2009 May;69(9):3755–3763.

[214] Stupack DG, Teitz T, Potter MD, Mikolon D, Houghton PJ, Kidd VJ, et al. Potentiation of neuroblastoma metastasis by loss of caspase-8. Nature. 2006 Jan;439(7072):95–99.

[215] Cutress RI, Townsend PA, Sharp A, Maison A, Wood L, Lee R, et al. The nuclear BAG-1 isoform, BAG-1L, enhances oestrogen-dependent transcription. Oncogene. 2003 Aug;22(32):4973–4982.

[216] Krajewski S, Krajewska M, Turner BC, Pratt C, Howard B, Zapata JM, et al. Prognostic significance of apoptosis regulators in breast cancer. Endocrine-Related Cancer. 1999 Mar;6(1):29–40.

[217] Kikuchi R, Noguchi T, Takeno S, Funada Y, Moriyama H, Uchida Y. Nuclear BAG-1 expression reflects malignant potential in colorectal carcinomas. British Journal of Cancer. 2002 Nov;87(10):1136–1139.

[218] Tang SC, Beck J, Murphy S, Chernenko G, Robb D, Watson P, et al. BAG-1 expression correlates with Bcl-2, p53, differentiation, estrogen and progesterone receptors in inva-

sive breast carcinoma. Breast Cancer Research and Treatment. 2004 Apr;84(3):203–213.