COMPUTATIONAL METHODS FOR ELECTRONIC HEALTH RECORD-DRIVEN PHENOTYPING

By

Peggy L. DeLong Peissig

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Clinical Investigation)

at the

UNIVERSITY OF WISCONSIN-MADISON

2013

Date of final oral examination: 6/28/13

The dissertation is approved by the following members of the Final Oral Committee:
Charles David Page, Jr., Professor, Computer Science, Biostatistics and Medical Informatics
Eneida A. Mendonca, Associate Professor, School of Medicine and Public Health
Amy Trentham-Dietz, Associate Professor, Population Health Sciencies
David L. DeMets, Professor, Biostatistics and Medical Informatics
Murray Brilliant, Senior Scientist, Marshfield Clinic Research Foundation

To Tomas,

for supporting and sharing my dream

and

Jacob, Cortney, Whitney and Jedd

for making it all worthwhile.

ABSTRACT

Each year the National Institute of Health spends over 12 billion dollars on patient related medical research. Accurately classifying patients into categories representing disease, exposures, or other medical conditions important to a study is critical when conducting patient-related research. Without rigorous characterization of patients, also referred to as phenotyping, relationships between exposures and outcomes could not be assessed, thus leading to non-reproducible study results. Developing tools to extract information from the electronic health record (EHR) and methods that can augment a team's perspective or reasoning capabilities to improve the accuracy of a phenotyping model is the focus of this research. This thesis demonstrates that employing state-of-the-art computational methods makes it possible to accurately phenotype patients based entirely on data found within an EHR, even though the EHR data is not entered for that purpose. Three studies using the Marshfield Clinic EHR are described herein to support this research.

The first study used a multi-modal phenotyping approach to identify cataract patients for a genome-wide association study. Structured query data mining, natural language processing and optical character recognition where used to extract cataract attributes from the data warehouse, clinical narratives and image documents. Using these methods increased the yield of cataract attribute information 3-fold while maintaining a high degree of accuracy.

The second study demonstrates the use of relational machine learning as a computational approach for identifying unanticipated adverse drug reactions (ADEs). Matching and filtering methods adopted were applied to training examples to enhance relational learning for ADE detection.

The final study examines relational machine learning as a possible alternative for EHR-based phenotyping. Several innovations including identification of positive examples using ICD-9 codes and infusing negative examples with borderline positive examples were employed to minimize reference expert effort, time and even to some extent possible bias. The study found that relational learning performed significantly better than two popular decision tree learning algorithms for phenotyping when evaluating area under the receiver operator characteristic curve.

Findings from this research support my thesis that states: *Innovative use of computational methods makes it possible to more accurately characterize research subjects based on EHR data.*

ACKNOWLEDGEMENTS

I would like to thank and acknowledge the many people who have been instrumental in the completion of this thesis and supported me during this journey.

My advisor and mentor, David Page, using kindness and encouragement has guided me through this process and taught me the value of collaboration. David is a man of great intellect, yet has the unique ability to communicate complex concepts in a way that was easy for me to understand. David allowed me the flexibility to pursue my own research interest, of which I am grateful. He has demonstrated both patience and enthusiasm for what I have been able to accomplish. It has been both an honor and great privilege to work with David during this graduate work.

I wish to thank my thesis committee members: David DeMets, Amy Trentham-Dietz, Eneida Mendonca and Murray Brilliant. They have been a wonderful source for guidance and encouragement. David DeMets through his wisdom guided me to the Clinical Investigation program and encouraged me to pursue this degree. Amy Trentham-Dietz, Eneida Mendonca and Murray Brilliant provided encouragement and spent countless hours reviewing manuscripts and/or reading my thesis in detail to provide feedback that enabled me to improve it. I embrace what I have learned from them and hope that I can someday do the same for others as they have done for me.

I am very grateful for having so many wonderful influences throughout my career and with great pleasure I extend my gratitude and a special thanks to: Michael Caldwell, Vitor Santos Costa, Cathy McCarty, Justin Starren, Dick Berg, David Gustafson and Luke Rasmussen. I have

valued their encouragement, advice and perspectives over the past years. I am indebted to them and hope to pass on their wisdom.

Thanks to the Institute for Clinical and Translational Research for developing the Clinical Investigation degree program. This degree program allowed me to combine my interests from three distinct academic disciplines into a single degree and thesis. I would also like to thank Debora Treu for her help in navigating through the requirements of graduate school.

I wish to recognize Marshfield Clinic, Marshfield Clinic Research Foundation and Security Health Plan for their support in allowing flexible work schedules so that I could pursue this degree. My research was not possible without support of the following funding sources: eMERGE Network, funded by NHGRI under U01-HG-004608; NIGMS grant R01GM097618-01; NLM grant RO1LM011028-01 and Clinical and Translational Science Award program grant 9U54TR000021.

I would like to thank my children, Jacob, Cortney, Whitney and Jedd; my extended family children, Tolea and Brandon; my siblings and my extended family as they have provided me with motivation, encouragement and continuous kidding which prompted me on. I would also like to recognize the late William B DeLong, my father, who instilled a strong work ethic and taught me to do my best and not to quit.

Lastly, I am indebted to my husband, Tom. Without his never ending love and encouragement this would not have been possible.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

ADE	Adverse Drug Event
ALI	Acute Liver Injury
AUROC	Area Under the Receiver Operator Characteristic curve
BN	Bayesian Networks
CART	Classical and Regression Trees
CDM	Conventional Data Mining
CDR	Clinical Data Repository
CHF	Congestive Heart Failure
Cox2ibs	Cox2 Inhibitors
CPT®	Current Procedural Terminology
cTAKES	clinical Text Analysis and Knowledge Extraction System
C4.5 & C5.0	Decision tree learning algorithm developed by Quinlan
CUIs	Concept Unique Identifiers
DNA	Deoxyribonucleic acid
DR	Diabetic Retinopathy
DW	Data Warehouse
Dx	Diagnosis
FP	False Positive
GH	Group Health Research Institute
GWAS	Genome-Wide Association Study
EHR	Electronic Health Record
eMERGE	electronic Medical Record and Genomics Network
EHR	Electronic Medical Record
ICD-9CM	International Classification of Diseases, Ninth Revision, Clinical Modification
ID3	Interactive Dichotomizer 3
J48	Java implementation of decision tree classifier based on C4.5
ILP	Inductive Logic Programming

KMCI	Knowledge Map Concept Identifier
NIH	National Institute of Health
NU	Northwestern University
MedLEE	Medical Language Extraction and Encoding system
ML	Machine Learning
N-P-C	Nuclear sclerotic, posterior sub-capsular and cortical cataract subtypes
NS	Nuclear sclerotic
NEG	Negative examples
NLP	Natural Language Processing
NPV	Negative Predictive Value
i.i.d.	Independent and identically distributed data
OCR	Optical Character Recognition
PART	Rule based classifier for machine learning
PMRP	Personalized Medicine Research Project
POS	Positive examples
PPV	Positive Predictive Value
SNOMED-RT	Systematized Nomenclature of Medicine Reference Terminology
SRL	Statistical Relational Learning
SVM	Support Vector Machines
TIFF	Tagged image file format
UMLS	Unified Medical Language System
VU	Vanderbilt University
WEKA	Machine Learning software
XML	Extensible Markup Language

CHAPTER 1

1.1 INTRODUCTION

The National Institute of Health (NIH) spends billions of dollars each year on medical research activities. In 2012 alone, approximately $147 billion dollars was spent funding various types of disease and medical condition related research (NIH RCDC Funding, 2013). Almost half of the competitive medical research funded by the NIH involves human subjects, or patients (Zinner et al., 2009). Accurately classifying patients into categories representing disease, exposures, or other medical conditions important to a study is critical when conducting patient-related research. Without this rigorous classification, also referred to as phenotyping, relationships between exposures and outcomes cannot be accurately quantified, thus causing varying and non-reproducible results in some clinical and genetic studies (Bickeboller et al, 2003; Schulze et al, 2004; Gurwitz et al, 2010; Samuels et al, 2009; Wojczynski et al, 2008).

Due to the availability of electronic patient information, the Electronic Health Record (EHR)—also called the Electronic Medical Record (EMR)--is increasingly being used to identify and characterize patients for medical research. The EHR contains highly relational and inter-dependent biological, anatomical, physiological and behavioral observations and facts that represent a patient's phenotype. EHR-driven phenotyping, a process whereby patients are electronically categorized using EHR data, has become a popular and cost-effective strategy for identifying large numbers of research subjects (Kho et al, 2011; McCarty et al, 2011).

The EHR-driven phenotyping process is largely dependent on multiple iterations of selecting patients and then manually reviewing them to identify classification criteria that can be programmed to select patients from the EHR. The process relies on the perceptions and knowledge of a multi-disciplinary team to uncover "hidden" relationships or "unseen" attributes found within the EHR data. As clinical experts (physicians) contribute to this effort, they describe attributes that are easy to see within their practice. They may miss attributes that they do not typically use when examining a patient but that are informative in the context of the EHR. Simply asking physicians what they want to search for is not optimal because, while they may "know it when they see it," they may not be able to anticipate ahead of time all the patterns in the EHR that will best correlate with a given disease. In addition, the probabilistic structures of EHR data are such that not all attributes that are observed are necessarily routinely recorded. Likewise, there may be other attributes that are recorded to substantiate billing or rule-out reasons that should not be considered necessarily true; for example, a diagnosis code of 410 (acute myocardial infarction), may be entered to justify billing for a troponin lab test to rule out myocardial infarction (MI), rather than being entered to categorically assert that the patient had an MI. Those sorts of correlations may not be visible to the physician and may lead to additional iterations of phenotype definition development. The result is a serious bottleneck in the construction of high quality phenotyping models.

The National Human Genome Research Institute has invested approximately $30+ million in the electronic Medical Records in Genomics (eMERGE) network to determine if EHRs can successfully be used to identify clinical populations for genome-wide association

study research (eMERGE Funding, 2011). One goal of that research network is to develop high throughput phenotyping methods that can be shared across institutions. High throughput phenotyping implies accelerating and expanding the current phenotyping process. Developing tools to extract information from the EHR and methods that can augment a team's perspective or reasoning capabilities to improve the accuracy of a phenotyping model or improve the efficiency of the phenotyping process is the focus of this research.

Because EHR-based phenotyping is important (Manolio et al, 2009; Ellsworth et al, 1999; Bickerboller et al, 2003; Schulze et al, 2004) and because phenotyping is hard and time consuming (Kullo et al, 2010; Wojczynski et al, 2008), conducting research on methods that improve the phenotyping process is critical to the advancement of medical research and the science surrounding EHR-driven phenotyping.

1.2 THESIS STATEMENT

Contemporary EHR-driven phenotyping methods have proven successful in producing high quality phenotypes. Nevertheless, phenotyping efforts I have been involved with or that have documented time and resources have typically run from six months to over a year and required numerous meetings and substantial time commitments from multiple clinicians, informaticists, programmers, and project coordinators. Much of that time is spent identifying attributes that accurately characterize the phenotype. My thesis is:

Innovative use of computational methods makes it possible to more accurately characterize research subjects based on EHR data.

1.3 CONTRIBUTIONS

With the increasing pressure to accelerate research and make it more efficient, there is a unique opportunity to advance the science surrounding EHR-based phenotyping and expand machine learning applications. This research contributes to both of these bodies of knowledge.

Electronic health record phenotyping and the application of computational methods based on machine learning approaches are general themes that connect the studies represented in this dissertation. Chapters 3-5 outline the ways in which this dissertation contributes to the fields of EHR-driven phenotyping and machine learning. Specifically these contributions are: 1) demonstrating the use of a multi-mode approach (structured queries, natural language processing and optical character recognition) to increase subject yield without compromising quality, for cataract and cataract subtype phenotyping; 2) using relational learning, particularly inductive logic programming (ILP), to identify adverse drug events in patient sub-groups, given the use of cox2-inhibitor medications; and 3) adapting ILP to the phenotyping task in a large relational database, by using unique cost functions and data censoring techniques; generating training sets without expert (physician) assistance for supervised machine learning; and infusing negative examples with borderline positive examples to improve ILP model performance.

1.4 DOCUMENT OVERVIEW:

The chapters of this document are organized as follows:

Chapter 2 is intended to provide basic background information that will be used to help the reader understand materials that are presented in Chapters 3-5. There are three main topics

that are presented: 1) the electronic health record; 2) phenotyping; and 3) machine learning. The chapter is designed to provide an overview of these topics and present popular methods or approaches used for phenotyping.

Chapter 3 introduces an approach that uses multiple computational methods to gather information for phenotyping. The methods include structured queries that are used on coded EHR data; natural language processing which is used to extract concepts from textual documents; and optical character recognition, used for identifying notations or characters on image documents; the latter two methods are based on machine learning as the underlying analytics to mine information. The chapter emphasizes the importance of using multiple methods to increase subject yield while still maintaining an adequate level of accuracy when phenotyping.

Chapter 4 introduces the relational machine learning approach and the novel methods of censoring by matching, temporal difference, and iterative interaction between a human and computer when developing models. Using medications (Cox2 inhibitors), I demonstrate the use of my methods to predict adverse drug events (ADE). An ADE is defined as a health-related problem or injury resulting from taking a normal dose of medication. This innovative ADE detection approach was then modified and applied to phenotyping in Chapter 5.

Chapter 5 introduces methods that use induction logic programming for the phenotyping task. It also presents an approach that minimizes physician involvement in the selection of training sets for the supervised learning activity.

Chapter 6 summarizes the contributions of this research and presents future areas of research.

CHAPTER 2

Background

This dissertation draws on two areas of research: EHR-driven phenotyping and computational methods such as machine learning. This chapter provides information on both subjects in addition to a detailed description of the electronic health record (EHR).

2.1 ELECTRONIC HEALTH RECORDS

The EHR represents a record of a patient's health information over time, as generated through on-going interactions with a health care system. Data from the EHR is usually transferred to a data warehouse and stored in relational schemas (refer to figure 2.1). There are many challenges when using this data.

Figure 2.1: Electronic Medical Record Example

Patient ID	Gender	Birthdate
P1	M	3/22/63

Patient ID	Date	Physician	Symptoms	Diagnosis
P1	1/1/01	Smith	palpitations	hypoglycemic
P1	2/1/01	Jones	fever, aches	influenza

Patient ID	Date	Lab Test	Result
P1	1/1/01	blood glucose	42
P1	1/9/01	blood glucose	45

Patient ID	Date	Observation	Result
P1	1/1/01	Height	5'11
P2	1/9/01	BMI	34.5

Patient ID	Date Prescribed	Date Filled	Physician	Medication	Dose	Duration
P1	5/17/98	5/18/98	Jones	Prilosec	10mg	3 months

1. Data are stored in multiple tables, rather than in one table with one record per patient, thus making it more complex to link a patient's medical history.
2. There is usually missing and/or incomplete information surrounding a patient's clinical history.
3. Patients may have unsubstantiated presumptive diagnoses in their medical record. For example, in some cases an ICD-9 diagnosis code is linked to an explanation that laboratory tests are being done in order to confirm or eliminate the coded diagnosis, rather than to define that a patient has the diagnosis.
4. Information found within the EHR is not always stored in a readily computable format (scanned images of hand written notes or test results, electronic text documents, etc.).
5. There is a lack of standardized entries in the EHR.
6. Clinical applications that support single data entry with minimal error checking result in erroneous data.
7. Clinical problems drive the EHR documentation practices, meaning there is poor negation of a disease or conditions when compared to reality. This point is critical when defining the control population because the "Absence of evidence is not evidence of absence" (Sagen, 2013).
8. There are methodological issues to address when using longitudinal data, e.g. multiple records for a given condition and multiple measurements – one must determine which ones to use (Wojczynski et al, 2008; Elkin et al, 2010).

Although there are known limitations of using EHR data there are also benefits for research. The EHR captures a variety of longitudinal information about a patient, ranging from detailed measurements to impressions provided by clinical experts. The EHR data is readily available for research, thus reducing the cost and time required to gather the data (Elkin et al, 2010). Traditional research data retrieval techniques usually capture data at defined point(s) in time and are not reflective of routine patient care measurements. In addition, the EHR data reflects ongoing interactions with the health system and also spans the continuum of health care representing primary, secondary and tertiary care events. While EHR data is often criticized for inaccuracies, (Hripcsak et al, 2011; Botsis et al, 2010) the reality is that the data is used in clinical care and it can be used for research (Herzig, 2009; Elixhauser, 1998; Peissig, 2012). Consequently, any genetic or clinical discovery translation into clinical practice must leverage the EHR data.

2.1.1 Data classes available within the EHR

The EHR retrieves and stores a variety of patient related data and information from a Clinical Data Repository (CDR). The CDR is optimized for patient care data delivery and data is returned usually in milliseconds, to clinical staff caring for the patient. There are three primary types of data found within the CDR.

- Structured or coded data – Demographic data such as name, address, gender, date of birth and death are usually stored as structured data. ICD-9-CM diagnostic and procedure codes and Current Procedural Terminology (CPT) procedure codes are administrative

types of data that are used for billing purposes. Laboratory results, clinical observations (such as height, weight, biometric data), and medication inventories and prescriptions are also structured types of data that are often stored with code, name, description and version attributes in a relational database (Denny, 2012). Structured data within the EHR is limited due to the effort required to capture it. Structured or coded data is almost always used to some degree, in the phenotyping effort.

- Narrative documentation – Narrative and semi-structured textual documents store transcribed notes, summaries of care such as hospital discharge summaries, clinical visit and telephone notes, interpretations from radiology exams and interpretations, laboratory interpretations and test results such as echocardiograms, angioplasty and surgeries, to name a few. There is a wealth of information embedded within clinical narratives and use within the EHR usually requires reading a document to gain information about a patient. Xu *et al*. (Xu et al, 2011) showed the use of clinical narratives identified colorectal cancer cases more reliably than coded EHR data such as ICD-9 and CPT codes. Clinical narratives are increasingly being used in the phenotyping effort. The volume of documents available in the EHR varies based on the length of time the EHR has been used for clinical care. These narratives
- Images and digital films – Clinical images including radiology, digital procedures such as an angioplasty procedure, genetic DNA scans, ophthalmology drawings are just a few examples of clinical images. Hand-written documents are often scanned as images into the EHR. Information retrieval from these types of media is usually manual and requires

reading or viewing and then interpreting by the reviewer. This media is rarely used when phenotyping because of the expense involved in training software to retrieve the data even though there is usually more image documents than clinical narratives and structured data combined.

2.2 PHENOTYPING

Careful phenotyping is both critical to the eventual results discovered by a study (Bickeboller et al, 2003) and a source of great challenge due to the variety of phenotyping approaches that can be employed with the same data (Schulze et al, 2004). Care must be taken at multiple levels with rigorous attention to data quality, completeness, comparability and recognition and reduction of clinical heterogeneity (Schulze et al, 2004; Wojczynski et al, 2008). This can be demonstrated by using a characteristic such as blood pressure. A simple approach distinguishes people who have had an elevated blood pressure based on one measurement found in a database containing blood pressure measurements from those who have not. However, a more useful approach would be to determine the extent to which a participant's average systolic (or diastolic) blood pressure over time is above or below their own expected average, given their age and body mass index at the time of each measurement. In order to generate a particular blood pressure phenotype, data on systolic and diastolic values, date of each measurement, date of birth (or age at measurement), weight and height at time of each measurement, plus data on the nature of the blood pressure reading, if available (sitting, standing, supine; on treatment or not) are needed. The phenotype would then be based on the residuals from a regression model

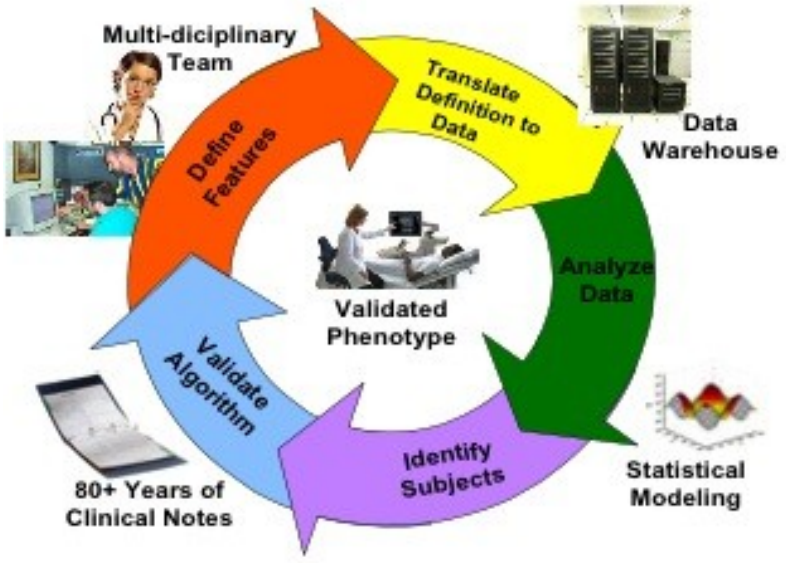
that attempts to predict mean blood pressure with mean age and mean body mass index. This method employs the available longitudinal data that can be found in the EHR and also adjusts for other contributors to blood pressure status such as blood pressure medication. These issues are particularly challenging to address when using data found within the EHR. Phenotypes are sometimes difficult to define because the disease or clinical manifestation may have an ambiguous or imprecise definition or be inadequately measured; the phenotype may encompass several underlying conditions where each have their own influences from genetics and the environment; or there may be unknown environmental or genetic influences (Wojczynski et al, 2008).

2.2.1 EHR Phenotyping Process

Defining a phenotype model representing a true disease state as well as intermediate phenotypes based on biological markers or clinical test results requires a clear understanding of methods to reduce clinical heterogeneity and/or deal with data quality issues. The current state of EHR-based phenotyping is maturing, but often a semi-manual process is used to derive valid phenotypes (see figure 2.2). A multi- disciplinary team identifies and then translates features representing subjects with the desired phenotype into a programmable data definition used to select patients from a data repository. In some instances, statistical modeling is done to reduce heterogeneity (Waudby et al, 2011). The data is analyzed and adjustments made to the model prior to pulling a sample of subjects to validate. Iterations of code revisions, record re-extraction and clinician review (validation) are usually required to increase the accuracy of the model.

2.2.2 Multi-disciplinary Phenotyping Teams

Figure 2.2: Phenotyping Process



Given the complex nature of phenotyping, having involvement from individuals spanning a variety of disciplines (medicine, epidemiology, biostatistics, informatics, programming, medical record abstraction and research) is extremely important. There are several advantages of phenotyping using a multidisciplinary team: 1) team members can offer a wide range of expertise from their respective disciplines; 2) they offer different perspectives when dealing with problems; and 3) teams members can usually work on tasks in parallel thus speeding the needed work.

Team members because of their cognitive ability, respective discipline and experiences bring different skills, knowledge and perspectives to the process, which can affect phenotyping outcomes if not managed effectively. A study by Barrick *et al.*, evaluated team member ability and personality, to see how those attributes affect team effectiveness (Barrick et al, 1998). Their study showed that general mental ability, conscientiousness, agreeableness, extraversion and emotional stability supported team performance activities. This work somewhat corroborates a study by Woolley *et al*. that looked at team composition and analytic effectiveness (Woolley et

al, 2008). The authors noted that bringing members with strong task-relevant abilities together with experts (those who possess a higher level of knowledge or skill than the average person) yields a greater potential to influence analytic performance. Nevertheless, building such a team can also present several challenges such as: 1) team member status dynamics may cause lower status team members to give more credence to experts than they deserve, or experts may not be inclined to take seriously the views of other team members; and 2) the abilities and skills of others may not be well known to all members and thus not utilized. The authors recommend conducting collaborative planning activities to define how members engage in explicit discussions, work assignment and how contributions will be used. The study demonstrated that team abilities and composition planning more positively impacts analytic team performance than team composition or collaborative planning alone.

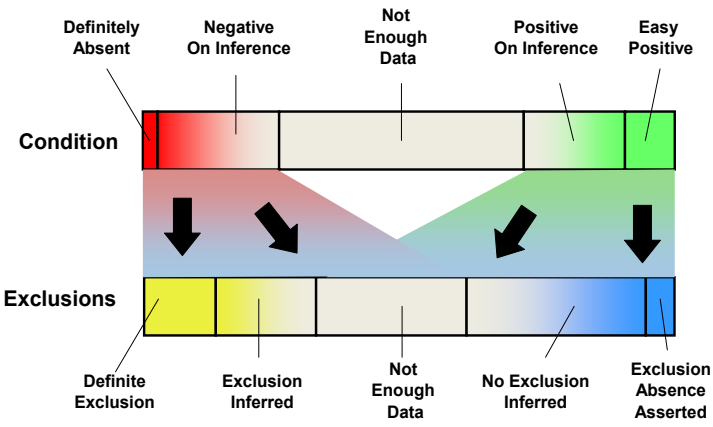
This research is relevant to the phenotyping process and team makeup because phenotyping is an analytic exercise at many levels. One person usually does not have the skills or knowledge to conduct all of the tasks required in the process. Bringing team members with different skills and cognitive abilities together to conduct phenotyping requires organization, planning and mutual respect of team member contributions. Recognizing the team dynamics, member abilities and providing a framework for work can improve the phenotyping process.

2.2.3 EHR Phenotyping Challenges

2.2.3.1 Phenotyping Tradeoffs

It is clear from the extent of articles published and creative activity surrounding the topic of EHR-based phenotyping for genetic studies (Kho et al, 2011; Wojczynski et al, 2008) that there is not, and probably cannot be, just one correct approach to phenotyping. A major challenge in the phenotyping process is to address tradeoff decisions between participant eligibility and phenotype homogeneity. In other words, the phenotype definitions that will succinctly characterize a subject and reduce heterogeneity will often require inclusion of more phenotypic factors with greater specificity in their definition, thereby reducing the total number of participants with the necessary data to be considered eligible for accurate phenotyping. The desired phenotype may require such specific data that many participants will be ineligible for a study. In order to phenotype a greater portion of participants, some relaxation of eligibility rules and data comparability may be required. Figure 2.3 illustrates this concept. As with the phenotype conditions, there are varying degrees of exclusions, and evaluation is needed to determine the tradeoffs.

Figure 2.3: Ins and Outs of EHR Phenotyping
Obtained from eMERGE Network (eMERGE, 2012)



The tradeoff concept can best be demonstrated using a cataract phenotyping example. Table 2.1 provides a "simplified" version of cataract criteria developed for EHR-driven phenotyping. Let's say that someone is conducting a genome-wide association study and

requires 1000 patients with a cataract (or who previously had a cataract) and 1000 patients who did not develop the condition. Prior to selecting patients for each group, an acceptable inclusion accuracy threshold (for selecting patients based on criteria or combinations of criteria) will be determined. Initially all criteria listed in table 2.1 could be used to identify probable cataract patients. One could speculate that a high percentage of patients identified using all of the criteria would actually have cataracts (denoted as "easy positives" in figure 2.3). This could be verified by reviewing the surgical and ophthalmology notes for the patients. What if only 800 patients were identified using all of the criteria? We still need 200 more patients for the study. The next step will be to relax the phenotyping criteria and use a subset of the criteria to see if there are patients (not originally selected) who have a cataract surgical code and not a cataract diagnosis code. If patients are identified, a similar verification of patient cataract status would ensue. Using surgical procedure codes for cataract patient detection is highly predictive of cataracts because a physician would bill for the cataract removal and would not want to miss a revenue generating opportunity. It is more difficult to identify cataract patients if the removal was done outside of the health care system because there would be no record of the event and no incentive to record this information. Thus far this example illustrates a trade-off on the condition continuum by relaxing specific eligibility criteria (which may add bias to the study), to gain more subjects for a study.

To continue this example, we next use the diagnostic code (without a CPT procedure code) to identify cataract patients. As before we select a group of patients not previously classified using diagnosis codes and verify a sample to estimate the percentage of patients who

actually meet the criteria. If the true positive rate meets the accuracy threshold, we will add these subjects to the cataract group. Note that subjects are becoming more like the controls because we are relaxing our criteria and identifying patients using different criteria (and also moving into the green/grey area on the continuum).

Table 2.1: Cataract phenotype algorithm description. This is a simplified version of a model used to identify patients with cataracts.

Criteria	Describing	Criteria	Description
1	Condition	Must have 1 Cataract removal surgery code	**Select the following CPT codes** '66982', '66983', '66984', '66985', '66986','66830', '66840', '66850', '66852', '66920', '66930', '66940'.
	Exclusion	Exclude traumatic, congenital and juvenile cataract surgery codes.	
	Condition	Must be age 50 or older at the time of surgery	
2	Condition	Must have 2+ cataract ICD-9 diagnosis codes	Senile Cataract 366.10 – 366.8 Unspecified Cataract 366.9
	Exclusion	Do not include any of these ICD-9 diagnosis codes	Congenital Cataract 743.30-.34 Traumatic Cataract 366.20 Juvenile Cataract 366.00-.11
	Condition	Must be age 50 or older at the time of surgery	
Control criteria	Condition	Absence of cataract procedure codes	See codes above
		Absence of ICD-9 Codes	See codes above
		Optical exam in last 2 years	

Identifying patients without a cataract is more difficult. One could logically assume that subjects without a surgical procedure or cataract diagnosis codes should be classified as controls. As previously noted data found within the EHR has limitations and must be evaluated prior to accepting it as ground truth. The control classification is dependent on several factors: 1) the longitudinal history of data found within the EHR or the length of time the patient has been cared for in the health system; 2) confirmatory tests; and/or 3) the workflow surrounding clinical data

collection. For example, a patient who had a routine optical exams and no indication of cataract diagnoses or procedures would likely be classified as a control, "Definitely absent" indicated in red on the condition continuum in figure 2.3. Another situation may involve a patient new to the health system who had a cataract previously removed. The patient will most likely not have documented cataract procedures and/or diagnoses because they are new to the system. The medical evaluation at the patient's initial visit may not include an eye exam, thus no fact would be recorded. In the last situation, a negation of cataract cannot be made without an eye exam, so we would place the patient in either the "Negative on inference" or "Not enough data classifications" on the continuum.

2.2.3.2 Feature Identification

In medicine, there is a distinction between anecdotal reports, syndromes (Syndrome, 2010), and disease. Anecdotal reports are generalized statements about something that is suspected for which no real evidence exists (anecdotal evidence). A syndrome implies the co-occurrence of several recognizable attributes (signs, symptoms or characteristics) that alert the clinician to the presence of other attributes (syndrome). A disease implies that we understand something about the causal process and the symptoms that occur. Often when a syndrome is first described, critical attributes are left out. Physicians describe attributes that are easy to see within their practice. As indicated previously, it is relatively easy for a physician to identify several clinical attributes that classify small numbers of patients as definitely having (or not having) a disease or phenotype. It is more difficult to identify hidden attributes (attributes that are

correlated with the initial attribute). This difficulty can lead to additional iterations when developing the phenotype model.

2.2.4 Phenotyping Tools and Technologies

Given the aforementioned challenges to EHR-driven phenotyping, the utility of using the EHR for phenotyping has been demonstrated in large-scale genomic studies (Ritchie et al, 2010; Peissig et al, 2012; Pakhomov & Weston, 2007; Kho et al, 2011). A variety of tools and technologies have been developed to support using the EHR for research and phenotyping activities. Following is a description of the major technologies that have become popular in the past two decades.

2.2.4.1 Data Warehouse

A well-accepted axiom in informatics is that it is difficult, if not impossible, for a single database to perform optimally for both single patient queries (retrieving all the data for one patient) and cross-patient queries (such as finding all patients with a diabetes diagnosis that have had an elevated HgbA1c lab result within the past year). A clinical data repository supporting an EHR is optimized for single patient queries and a data warehouse is optimized for cross patient queries. The data warehouse is an extension of the EHR that combines data from the clinical data repository and other disparate clinical databases into an integrated repository that is maximized for population-based queries. Data stored within the data warehouse is usually structured (or coded), standardized, time variant and relational in structure. The efficiency of phenotypic queries is generally increased by the availability of a data warehouse.

2.2.4.2 Structured Queries and Data Manipulation

Notably the most popular approach to phenotyping, structured queries take advantage of structured data found within the DW. Structured queries use Boolean logic and programmable rules to identify patients with a given phenotype. The basic Boolean logic can be embedded into computer programs using a variety of programming languages that allow interaction against tables in the DW. Structured queries require less time and effort (in most situations), and are dependent on available coded data to classify patients. Advanced analytic and data management approaches filter, transform and graphically present data in ways that allow humans to determine phenotyping parameters. A drawback of this approach is the person writing the structured queries has to write the code to pull the information and thus must know the criteria and structures of the database. Many of the EHR-driven phenotyping efforts take advantage of these techniques as a way to characterize patients.

2.2.4.3 Natural Language Processing

Clinical documents found within an EHR are valuable sources of information for phenotyping (Goryachev et al, 2006). In order to use information embedded in the textual documents, Natural Language Processing (NLP) is used to transform unstructured text data from the documents into a structured format that can be used for phenotyping. Several NLP approaches and systems have been developed to extract concepts, measurements or clinical information on a variety of diseases and conditions. Some of the more popular NLP systems are:

- Medical Language Extraction and Encoding (MedLEE) - a system that was developed at Columbia University by Friedman et al. (Friedman et al, 1994, 1995), uses syntactic and semantic parsing of data. This linguistic rule based system was originally designed for radiology reports (Mendonca et al, 2005; Friedman et al, 1994, 1995) and has since been expanded to other clinical domains (Friedman et al, 2004; Melton and Hripcsak, 2005; Peissig et al, 2012).
- MetaMap - is a freely distributed NLP system that was originally designed to extract information from medical literature. It was developed by Aronson et al. at the National Library of Medicine and maps to concepts found in the Unified Medical Language System Metathesarus (Aronson, 2001; Lindberg, 1993).
- Knowledge Map Concept Identifier (KMCI) - a proprietary general purpose NLP system developed by Vanderbilt University (Denny, 2003, 2005, 2009). It supports concept identification and negation and is used for a variety of phenotyping initiatives to support genome-wide association studies.
- clinical Text Analysis and Knowledge Extraction System (cTAKES) - Savova et al. from Mayo Clinic developed cTAKES. cTAKES is an open source NLP system and consists of several modules placed into a pipeline architecture (Savova et al, 2010).

Already there is a large body of research surrounding NLP and the ability to extract clinical concepts and data from text-based clinical notes (Friedman et al, 2004, Denny et al, 2003, 2004, 2009; Peissig et al, 2007; Mendonca et al, 2005). NLP has contributed significantly

to both the accuracy and efficiency of developing EHR-based phenotyping algorithms (Pakhomov et al, 2007; Li et al, 2008; Peissig et al, 2012).

Several studies have compared the accuracy of phenotypes developed using administrative coded data to phenotypes derived using NLP methods. Pakhomov *et al.* found NLP to be superior over diagnostic coding to detect patients with angina pectoris (chest pain) (Pakhomov et al, 2007). Li *et al*. compared NLP to ICD-9 coding for extracting screening information from discharge summaries (Li et al, 2008). Several advantages were noted when using NLP, but the authors indicated that more study was needed. Elkin *et al*. used Mayo Clinic's Vocabulary Parser to code SNOMED-RT diagnoses from clinical documents and found the Vocabulary Parser to be significantly better than ICD-9 coding when evaluating 10 diagnostic conditions (Elkin et al, 2001). A study by Saria *et al*. demonstrated that combining EHR coded data with a natural language processing approach boosts model accuracy significantly over the language approach alone, when identifying complications of pre-mature infants (Saria et al, 2010). Although these studies have indicated the superiority of NLP over structured query approaches one must evaluate the phenotype and EHR to determine the best approach for phenotyping. The approach may vary depending on phenotype and EHR.

2.2.4.4 Optical Character Recognition

The use of optical character recognition (OCR) technology provides the ability to extract clinical information from image documents. Often figures or documents are scanned into the EHR that contain clinically relevant information for research. OCR technology is used to

interpret characters and/or symbols found within the document. The software is trained to recognize characters or symbols and then translates the characters/symbols into usable data. Kim and Yu developed an OCR tool to extract data from biomedical literature figures (Kim, 2011). Rasmussen et al. applied OCR to identify cataract subtypes phenotypes in patients for genome-wide association study phenotyping (Rasmussen, 2011; Peissig, 2012).

2.3 MACHINE LEARNING

Machine learning (ML) is a computational discipline aimed at creating algorithms that allow computers to use large amounts of data to build predictive models or to recognize complex patterns or characteristics within data. The advantage of using ML approaches is that computers don't have to be programmed with the explicit patterns in advance, but can find the most informative patterns by examining the data.

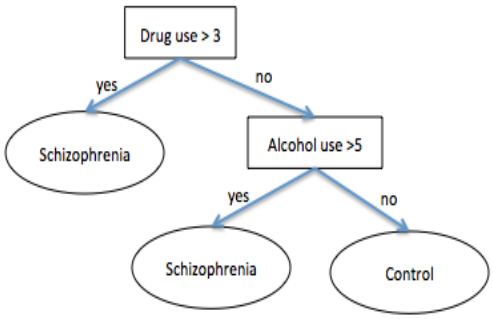
Supervised learning is one type of machine learning that learns patterns (or hypotheses) between inputs and outputs to predict future outcomes or values (Getoor and Taskar, 2007). The supervision comes from a training set of examples that are labeled with an attribute of interest that we would like to predict from the other attributes. This attribute is typically called the response variable or response; if it is nominal, it may also be called the class. A hypothesis, or predictive model, is constructed from a space of predictors. Over time, many different supervised learning approaches have been developed and applied to the health care domain. The NLP and OCR phenotyping methods previously reported rely on supervised learning methods. Mitchell provides a general overview of approaches used for machine learning (Mitchell, 1997).

These methods are based on the inductive learning hypothesis which states that any hypothesis found to accurately label a large set of observed training examples is also likely to provide accurate labeling when applied to unobserved data. In the following section I provide an overview of the machine learning methods relevant to this research.

2.3.1 Decision Trees

Decision tree learning is similar to a series of if-then-else statements or a flow chart. The learned model is represented by a decision tree (figure 2.4). Each example is classified by starting at the root node of the tree. Starting with the top node of the tree as the current node, the attribute test at that node is applied to the example by comparing the test value in the node with the attribute value in the example. One branch from the current node is consistent with the attribute value for the example, and we proceed down that branch. The end of a branch is represented by a leaf, which provides the predicted value for the response variable.

Figure 2.4: Decision Tree Example
(Example from Struyf et al, 2008)



Decision trees are used for classification or prediction. They are easy to understand, implement and used to visually and explicitly represent decisions. Decision trees can be divided into two main types: 1) a classification tree when the result of the learning activity results in a prediction of a classification (i.e. predicting if a patient will have a heart attack); and 2) a

regression tree which results in a prediction of some type of value (i.e. predicting the amount or level of coumadin needed to thin ones blood after a hip replacement surgery to prevent blood clots). Construction of a decision tree from data, or training of a decision tree, proceeds by recursively partitioning the data based on some scoring function for the purity of a node, such as information gain or Gini gain. The next section discusses this process in greater detail for the specific algorithm ID3, which uses information gain. Gini gain will be described later in this chapter with Classification and Regression Trees.

2.3.1.1 Interactive Dichotomizer 3

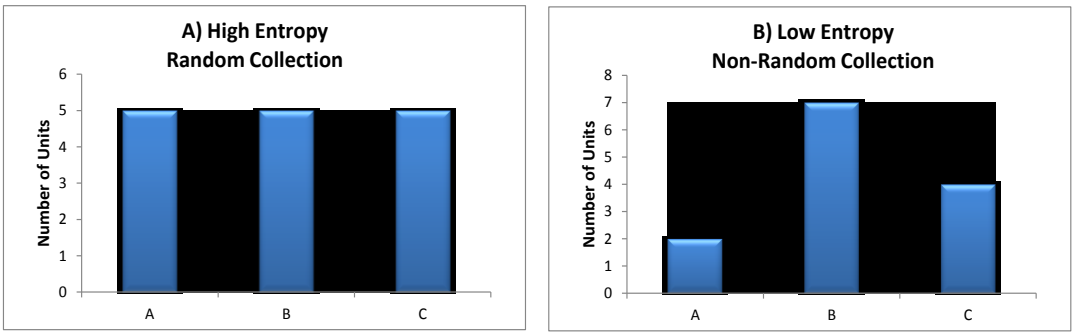
A popular decision tree algorithm is Interactive Dichotomizer 3 (ID3) (Quinlan, 1986) and successors C4.5 and C5.0. These algorithms employ top-down classification where an instance is classified by starting at the root node of the tree. Each attribute is tested and then continues down the branch if the value corresponds to a node value. The process is repeated for the sub-tree rooted at the new node. A greedy search is used to search the space of possible decision trees meaning that the algorithm does not backtrack to consider earlier choices once a candidate attribute is chosen. ID3 uses an information gain measure to select from candidate attributes at each step while growing the tree. Information gain is a measure of difference between two probability distributions. Simply put, information gain is the expected reduction in entropy caused by knowing the value of the attribute. Information gain is calculated using the following formula:

$$\mathrm{IG}\ \mathrm{X}\ \mathrm{Y}) = H\ (\mathrm{X}) - H\ (\mathrm{X}\,|Y)$$

where: IG denotes information gain
H denotes information entropy (definition follows)
Y denotes an attribute, or variable
X denotes the class (response) variable

Information gain is based on information theory concept of entropy (Shannon, 1948), which measures the uncertainty in a random variable (Mitchell, 1997). For example, suppose there is a variable that has three values: A, B, and C. If the collection of values for the variable occurs randomly (see figure 2.5.A), the variable is said to have high entropy and each value has an equal chance of being selected or used. If on the other hand, the collection of values are non-randomly distributed (see figure 2.5.B), the variable is said to have low entropy. In this

Figure 2.5: High and Low Entropy



situation, value B occurs more frequently than A or C.

In figure 2.5.A above, the entropy is closer to 1 (because of randomness) and the values sampled from it would be roughly evenly distributed. Figure 2.5.B, entropy is closer to 0 and the values sampled would be more predictable with the selection of B often (Moore, 2003). Entropy is calculated using the following formula:

$H(X) = (p1 \log2 p1) – (p2 \log2 p2) - \ldots pm \log2 pm$
$= \sum_{j=1}^{m} p_j \log2 p_j$

where H(X) is the entropy of X; and p is the probability; (Moore, 2003)

An advantage of ID3, and decision trees in general, is that they can learn non-linear classifiers. Another is that its greedy search makes it computationally efficient. A limitation of ID3 is it does not perform backtracking in its search. Once it selects an attribute to test, it does not reconsider its choice and thus could converge to a locally optimal solution that may not be globally optimal. A related limitation is that ID3 only maintains a single current hypothesis. By doing so it loses the capability to explicitly represent all consistent hypotheses. ID3 also operates using selection bias for trees that place attributes with highest information gain closest to the root, which in turn favors shorter trees over longer ones. When using ID3 and other decision tree algorithms, caution should be taken to avoid over-fitting of the tree to the training data. Pruning the tree may be needed to make algorithms more generalizable.

2.3.1.2 C4.5

One of the best-known and most widely used learning algorithms is C4.5, which is an improved version of Quinlan's ID3 algorithm. A limitation of ID3 is that it favors attributes with large numbers of divisions, which leads to over fitting. The C4.5 algorithm is designed to overcome the disadvantages of information gain and is sensitive to how broadly and uniformly the attribute splits the data (Quinlan, 1993). The C4.5 algorithm handles continuous data and can deal sensibly with missing values by treating it as a separate value. It also has capabilities to prune a tree when using noisy data. It also has the capability to develop rules by greedily pruning conditions from each rule if it reduces the estimator error of the training data. A

disadvantage of using C4.5 is that it is computationally slow when using large and noisy datasets. A commercial version C.50 uses a similar technique but is much faster.

2.3.1.3 Classification and Regression Trees

Classification And Regression Tree (CART) rule-based classifier is a non-parametric decision tree learning technique that is used to create decision lists (sets of rules) (Breiman, 1984). CART builds classification or regression trees for numeric attributes (regression) or categorical attributes (classification). The algorithm will identify the most significant variables by forming a partial decision tree and turns the "best" leaf into a rule with each iteration. At each node, the available attributes are evaluated on the basis of separating the classes of training examples. The tree building process entails finding the best initial spit at the root node. A popular splitting function used by CART is the Gini index, which is described in the next paragraph. For each sub-node, we find the best split for the data subset at the node and continue this until no more splits are found. We then prune the nodes to maximize generalizability.

The CART algorithm uses the Gini index (also referred to as Gini coefficient or Gini impurity), to determine how to evaluate splits. The Gini index is a measure how often a randomly selected element from a set would be incorrectly labeled if it were labeled using the distribution of labels in the subset. A best split is the one that maximizes the purity (a single class is primarily represented) for an attribute. The Gini index (Breiman et al, 1948) is calculated as follows:

$$gini\ \ C\ \ = 1 - \ \ \sum_{j=1}^{n} p_j^2$$

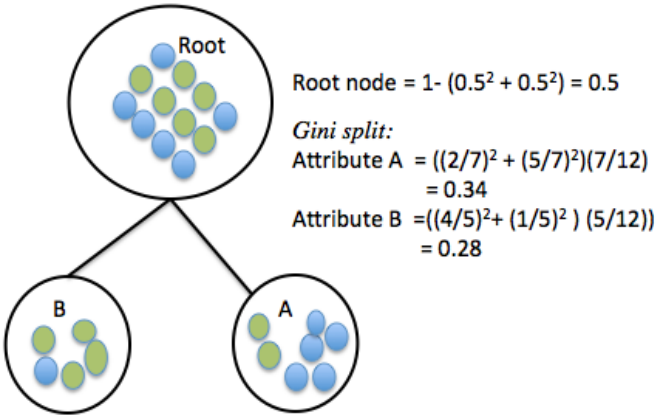
where pj is the relative frequency in class C.

In figure 6, we see that the root node (represented by a variable with 2 classes of data), has a Gini index of 0.5. Using the equation above, we would calculate the index in the following way: $1 - \left(\frac{6\ green}{12} \right)^2 + \left(\frac{6\ blue}{12} \right)^2 = 0.5$. Once the splitting of C occurs into two subsets C1 and C2 with a size of N1 and N2, the Gini index for the split is calculated using the this equation:

$$gini\ split\ \ C\ \ = \frac{N1}{N}\ gini\ \ C1\ \ + \frac{N2}{N}\ gini\ (C2)$$

This equation considers the number of elements in the sub-node and weights the Gini index calculated for each sub-node by the sub-node weight (the total elements for the sub-node / total number of elements represented by all sub-nodes). The result is an index that can be used

Figure 2.6: Gini Splitting at Root and Sub-nodes



to select the split. Figure 2.6 illustrates the calculation of the Gini index for the root node and Gini split. In this example attribute B (smallest Gini split (C)) is chosen to split the node (Rokach and Maimon, 2010).

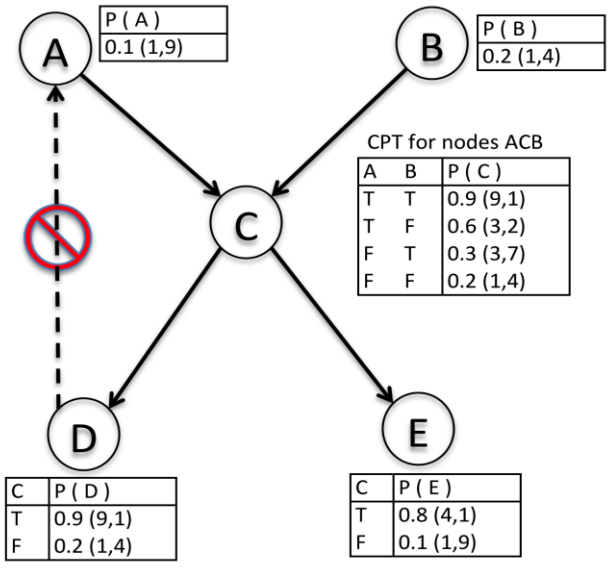
2.3.2 Association Rule Learning

Association rule learning is based on the premise that given a set of transactions, find the rules that will predict the occurrence of an item based on the occurrence of other items in the transaction. This approach is commonly used in Market-basket transactions and implies co-occurrence not causality. The mining task goal is to list all possible association rules and then compute the support and confidence for each rule and prune rules that fail to meet the minimum support and confidence thresholds. This approach is computationally prohibitive and thus the number of candidates, transactions or comparisons must be limited (Liu, Hsu and Ma, 1998).

2.3.3 Bayesian Networks

A Bayes network (BN) is a graphical structure that allows us to easily understand and reason about an uncertain domain (Mitchell, 1997). The nodes of a Bayesian network represent random variables (denoted in figure 2.7 as A, B, C, D, and E) and the lines between the nodes are directed arcs or links that represent dependencies between the variables. These directed arcs depict the direction of the relationship between a parent node (e.g. A) and child node (e.g. C). The strength of the

Figure 2.7: Example of Bayesian Network and Conditional Probability Tables



Example from: © Jude Shavik, 2006 & David Page 2007

relationship between the nodes is quantified using a conditional probability table (CPT). The only constraint surrounding the Bayesian network is that there cannot be any directed cycles. In figure 2.7 there is a directed cycle between nodes A, C and D (represented by the dashed line). One cannot return to a node simply by following the directed arcs as shown with arc DA. Another assumption is based on the Markov property and implies that there are no direct dependencies in the system being modeled that are not shown via arcs. Bayesian networks having this property are called independence-maps. In the example provided in figure 2.7, the joint distribution table for nodes A, C, and B is represented by the "CPT for nodes ACB". This table provides probabilities for C based on the combination of values that A and B contribute. For example, the probability of C given A and B are both true is 0.9. If new information is obtained, we update the distribution based on relevant data points that correspond to the CPT entry.

A Bayesian network specifies the full joint distribution over its random variables. The nodes are represented using x1 to xn and the value in the joint distribution is represented by P(x_1, x_2, … xn), where P is the product of the joint distributions. The chain rule of the probability theory (Russell and Norvig, 2003), allows us to show the basic equation as:

$$P\ x_1, \ldots x_n\ = (\ \prod_{i=1}^{n} P(x_i | Parents\ X_i\ )$$

To create a Bayesian network, we first have to identify the random variables in the network and capture the relationships between the nodes using arcs, as illustrated in figure 2.7. Each random variable in the BN has values that are both mutually exclusive and exhaustive. The variables can be Boolean nodes (true or false); ordered values (cold, warm, hot); or integral

values (weight = 20kg – 200kg). Using a simplified version of Pearl's Network Construction Algorithm (Pearl, 1988), the following steps occur: 1) choose a set of relevant variables for {Xi} where Xi represents a random variable or node that describe the domain; 2) order the variables <X1, … Xn>; 3) Start with an empty network and add variables one at a time until there are no variables; For each variable Xi ; add the arcs to the Xi node from already existing nodes in the network (Parents of Xi) and satisfy the conditional independence property; then define the CPT for Xi. The conditional independence property is:

$$P\ X_i\ X_i', \ldots, X_m') = P(X_i \mid Parents\ X_i\ )$$

where X_i' *...* X_m' *are all variables that precede* X_i

For each node in the network we have to identify a complete setting for all variables. We also assume that the data set we are using is a random sample from the distribution that we're trying to model. If available, we can use a prior distribution and simply update the distribution based on the relevant data points (that agree with the settings for the parents that correspond with the CPT entry.) This is referred to as a Dirichlet distribution (Geiger and Heckman, 1995). Stated simply a Dirichlet distribution is a distribution over a distribution.

One challenge when using Bayesian networks is to represent hidden variables. These variables may affect the Bayesian network, but because they are hidden, cannot explicitly be measured and thus not included in the network (Page, 2013). We may want to include a node in the network to represent this phenomenon. Other variables can be used to predict the hidden variable and the hidden variable can also be used to predict other variables. Trying to estimate CPTs for this is difficult because none of our data points have a value for this variable. The

general EM framework addresses this problem and is used to estimate CPTs for hidden variables (Page, 2013). The first step in the EM framework is to calculate the expectation (E), over the missing values for the given model. The second step is maximization (M), which replaces the current model with a Dirichlet distribution that maximizes the probability of the data.

There are situations where calculating missing or sparse information is resource intensive and the number of structures created is exponential. Finding an optimal structure is NP-complete, meaning there is no known efficient way to locate a solution. Two common options are used in this situation: 1) severely restrict the possible structures and use Tree-Augmented Naïve Bayes (Friedman, Geiger and Goldszmidt, 1997); or 2) use a heuristic search (such as sparse candidate) (Friedman, Nachman and Pe'er, 1999).

2.3.4 Relational Learning

Over the last decade, ILP and other methods for relational learning (Getoor and Taskar, 2007) have emerged within the ML domain to address the complexities of multi-relational data. These relational learning methods have been used with EHR data in studies ranging from screening for breast cancer (Burnside et al, 2009; Liu et al, 2012) to predicting adverse drug events (Davis et al, 2008; Weiss et al, 2012) or adverse clinical outcomes (Page et al, 2012; Berg et al, 2010; Kawaler et al, 2012; Davis et al, 2012).

Unlike rule induction and other machine learning algorithms that assume each example is a feature vector, or a record, ILP algorithms work directly on data distributed over different tables for diagnoses, labs, procedures, prescriptions, etc. ILP algorithms search for non-recursive

Datalog rules, equivalent to SQL queries or relational algebra expressions, that differentiates positive examples (e.g., cases) from negative examples (e.g., control patients) given background knowledge (e.g., EHR data). The algorithmic details of leading ILP systems have been thoroughly described (Dzeroski and Lavrac, 2001; Inductive logic programming). In a nutshell ILP uses a covering algorithm adopted from rule induction algorithms to construct a set of rules, known as "clauses". The covering algorithm starts with an empty set of clauses, or empty hypothesis, and searches for a clause that maximizes a score of positive vs. negative examples explained by the clause, adds the clause to the hypothesis, and removes the positive examples explained. These steps are repeated until all the positive examples have been explained.

Rules or clauses are constructed by starting from an unconditional rule, or empty clause, and adding antecedents to the rule one by one. New antecedents are generated by enumerating possible calls to the database. For example, ILP could enumerate the diagnosis codes reported in the database. Ideally, one would only look for clauses that explain positive examples and do not explain, or cover, any negative examples. In practice, ILP must deal with inconsistent and incomplete data hence it uses statistical criteria based on the number of positive and negative explained examples to quantify quality. Two simple criteria are to score clauses by the fraction of covered examples that are positive, which is precision, or by the number of positive examples minus the number of negative examples covered by the clause.

2.3.4.1 Inductive Logic Programming Explanation

Inductive Logic Programming (ILP) addresses the problem of learning (or inducing) first-order predicate calculus (FOPC) rules from a set of examples and a data-base that includes multiple relations (or tables). Most work in ILP limits itself to non-recursive Datalog (Ramakrishnan, 2003), a subset of FOPC equivalent to relational algebra or relational calculus. Consequently it builds upon the concept of an IF-THEN rule. IF-THEN rules are one of the most popular representations in data-mining and machine learning, and are of the form shown in this example:

IF Sunny AND Vacation THEN PlayOutside

This rule states that if it is sunny and vacation time, it is time to play outside. Observe that the rule implicitly refers to an individual. First-Order rules use variables to make it explicit the individuals, to which they refer to, making it possible to refer to different individuals. As an example, we use Mitchell's (Mitchell, 1997) Datalog rule for granddaughter:

IF Father(x,y) AND Mother(y,z) AND Female(z) THEN GrandDaugther(x,z)

x, y, and z are variables that can be set (or bound) to any person, but only the values consistent with the database will make the rule true. Notice that not only does this rule refer to multiple individuals, but that it also refers to multiple tables in the database, or predicates: Father, Mother, and Female. This ability to mention different individuals whose properties are spread over different tables, added to the fact that rules have an intuitive translation to natural language, makes Datalog rules a powerful and natural representation for multi-relational learning.

A large number of different learning algorithms have been proposed for learning rules within ILP (De Raedt, 2008). All these algorithms are designed to search for good rules, and they do so by constructing rules, evaluating the rules on the data, and selecting the rules that do well according to pre-defined criteria. The first widely-used algorithm, Quinlan's FOIL (Quinlan and Cameron-jones, 1995), executes by first generating all possible rules of size 1, then all rules of size 2, and so on until either it finds good rules, or it reaches some threshold and stops.

The problem with FOIL is that in most domains there are a large number of possible rules one can construct. For example, in a typical EHR, we may find over 5,000 different diagnoses codes, over 3,000 different medications, and thousands of different possible labs and procedures (Muggleton, 1995). Rules must refer to specific conditions, drugs, or labs and in this case, applying the FOIL procedure would generate at least 10,000 different rules of size 1, over 10,0002 clauses of size 2, and so on. Unfortunately, evaluating these rules over thousands or maybe millions of patients is not practical.

In this work I use Muggleton's Progol algorithm (Linder, Bates and Middleton, 2007), as implemented in Srinivasan's Aleph system (Ashwin, 2001). Progol improves on FOIL by applying the idea that if a rule is useful, it must explain (or cover) at least one example. Thus, instead of blindly generating rules, Progol first looks in detail at one example, and it only constructs rules that are guaranteed to cover that example. In other words, Progol still generates rules in the same fashion as FOIL but it uses an example, called a seed, to guide rule construction. The benefit is that instead of having to generate a rule for the thousands of

conditions, drugs and labs in the data-base, we can generate rules for the much lesser number of conditions that affect a patient.

2.3.4.2 Inductive Logic Programming Progol Algorithm

In more detail, the Progol Algorithm is as follows:

- Select an example not yet explained by any rule. In the EHR domain, an example is a patient's clinical history.
- Search the data-base for data directly related to the example. In the case of an EHR, this means collecting all diagnoses, prescriptions, lab results, for the selected patient.
- Generate rules based on the patient, using the FOIL algorithm. The rules will be constructed from the events of the chosen patient's history, but must also explain other patients. This is achieved by replacing the references to the actual patient and temporal information by variables. The procedure stops when it finds a good rule (according to the criteria I discuss later).

Remove the examples explained by the new rule. If no more examples remain, learning is complete. Otherwise, repeat the process on the remaining examples, starting from step 1.
ILP learning is thus somewhat different from learning with a propositional system or single table, as most ML algorithms do. Instead of using a single table, the first step must be to define which tables are of interest to the learning process. Notice that it is not necessary for tables to be materialized; implicit tables or views may also be used (Davis et al, 2005).

The second step is to parameterize the search. In the case of phenotyping, accepted rules should cover very few, ideally zero, negative examples. Next, rules that succeed on very few examples tend to over fit: a useful heuristic is that a rule is only acceptable when it covers at least 20 examples. Last, search time heavily depends on the maximum number of rules that are considered for each seed.

The actual search process is automatic. The output is a set of rules (or theory). Each rule will cover at least one example, and quite often more than one example. Notice that whether a rule covers an example or not, the rule may be seen as a property, or attribute (that is true or false), of the example.

If ILP learned rules can be seen as attributes, we can construct classifiers that combine the output of the rules. Any classifier can be used. In this work, we use the TAN Bayesian networks, an extension of naïve Bayes that better addresses correlated attributes, as we have had previous good results in using TAN for related tasks and it produces probabilities for examples being true (Davis et al, 2005).

2.3.4.3 Statistical Relational Learning

Statistical relational learning (SRL) combines graphical model approaches (denoting explicit models of uncertainty) with ILP to construct probabilistic models to analyze relational databases. The SRL approaches learn the joint probability distributions of fields in the relational database to predict disease outcomes and support noisy, uncertain and non-i.i.d. real world data (Muggleton, King and Sternberg, 1992). There are a variety of SRL approaches that address

EHR data issues such as: 1) missing and incomplete data; and 2) large amounts of data causing long run times. Natarajan *et al*. utilized probability distributions instead of binary responses like "true" or "false" when learning relationships among fields. This enabled SRL to quickly build classifiers that can easily track significant improvements in the prediction algorithm (Getoor and Taskar, 2007).

2.4 EHR CHALLENGES WITH MACHINE LEARNING

The data from EHRs pose significant challenges for classical machine learning and the data mining approaches (Getoor and Taskar, 2007; Page et al, 2012). First, there are millions of data points represented for each patient within the EHR (Linder et al, 2007). Knowing which facts to use and how they relate often requires clinical intuition. Second, EHR data has multiple meanings. For example, in some cases an ICD-9 diagnosis code is linked to an explanation that laboratory tests are being done in order to confirm or eliminate the coded diagnosis, rather than to define that the patient has diagnosis. Third, there is missing measurement data. Finally, an EHR is multi-relational, and classical machine learning methods require "flattening" of the data into a single table. Known flattening techniques, such as computing summary features or performing a database join operation could result in loss of information (Getoor and Taskar, 2007).

2.5 MACHINE LEARNING AND PHENOTYPING

One possible approach to constructing high quality phenotype definitions is to apply the mathematical discipline of Machine Learning. Machine Learning is aimed at designing and creating algorithms that allow computers to develop behaviors based on use of empirical data, utilize large amounts of data to build predictive models or to recognize complex patterns. Machine learning has been used successfully in a variety of non-health care domains to discover relationships within large databases (data mining) and to provide insight when humans did not have enough knowledge of the data to develop effective predictive models of discovery. Machine learning has also been used in the health care domain in the context of natural language processing to pull concepts and information from textual documents (Roberts et al, 2009) and for genomic discoveries.

The Machine learning literature is predominantly filled with research highlighting the design and development of machine learning algorithms. The empirical results of this type of research aim to answer the question: Is algorithm "A" better than algorithm "B"? "Better" is usually measured in terms of accuracy and reported in terms of error rates; precision/recall; sensitivity/specificity; positive/negative predictive value; area under the ROC or precision/recall curve; F-score; or some statistical tests such as t-tests that show the differences based on cross validation goals or by bootstrapping.

Recently there has been interest in using machine learning or machine learning systems as a tool for EHR-based phenotyping to both improve the accuracy of the phenotypes and also reduce the time needed to develop the EHR-based algorithms. Because of the limited duration of

EHR implementations, research relating to EHR-based phenotyping is relatively new and there have not been any literature reviews describing the application of machine learning to the domain. The following section will examine the evidence regarding the application of machine learning algorithms to the EHR-based phenotyping process to evaluate if accuracy has improved or if time was reduced when compared to the traditional physician-lead phenotyping process.

2.5.1 Machine Learning Phenotyping Literature Review

In July of 2011, I conducted a literature review to evaluate the potential role of using ML techniques to improve the accuracy or reduce the time of EHR-driven phenotyping. Only studies that characterized subjects using coded data from the EHR were considered for this review. In addition, the studies had to involve some type of phenotyping activities using both machine learning algorithms and some comparison against the traditional physician-lead approach to phenotyping. The outcome of the comparison would be some measure of algorithm diagnostic accuracy (the ability to identify a phenotype correctly using the algorithm).

A total of 571 studies were screened for inclusion by reviewing the title. Of those, a total of 60 unique articles were selected for a more detailed abstract review. Thirty of the abstracts screened articles were eliminated because the content did not demonstrate EHR-based phenotyping efforts. These studies could be categorized as:

- Prospective, screening, prediction or prevalence studies – the focus was not on phenotyping (5, 16%)
- Recruiting or system evaluation studies (11, 36%)
- System or application development studies (5, 16%)
- Natural language processing only studies (3, 10%)

- Opinion or review papers (4, 13%)
- Other (3, 10%)

Of the remaining 30 studies, a total of six studies were identified as using ML approaches for phenotyping (Anand and Downs, 2010; Huang et al, 2007; Pakhomov et al, 2007; Wu et al, 2010; Xu et al, 2011; Carroll et al, 2011). Only two of the six ML studies compared the accuracy of the ML approach to the traditional phenotyping processes, although there were limitations with the evaluations (Pakhomov et al, 2007; Carroll et al, 2011).

There were considerable differences in ML approaches used, the phenotypes and approaches to validation. The ML methods used by the six studies were: Support Vector Machines (SVMs), logistic regression, Ripper, Naïve Bayes, IB1, C4.5, Noisy-OR and recursive and adaptive Noisy-OR. In addition, there were differences in phenotypes and outcome measures.

A study by Carroll *et al*. (Carroll et al, 2011) indicated that it was possible to create high performance algorithms (using support vector machines) when training on naïve and refined data sets. The algorithms significantly outperformed the traditional phenotyping approach. This study also showed that future machine learning algorithm development may be possible with only small numbers of manually identified cases (about 50-100 cases) thus indicating less time needed for algorithm development and validation. This limits the generalizability of the results. In addition, only one non-blinded physician created the gold standard for the investigation.

Pakhomov *et al*. (Pakhomov et al, 2007) conducted three phenotyping evaluations: 1) NLP with ICD9 diagnostic codes; 2) manual reviewed records with a ML-based approach; and 3)

NLP to manually reviewed records. A direct comparison between code-based phenotyping and ML-based methods was not done. This comparison was problematic for several reasons: the sample sizes and populations were different between each of the comparisons and the reference dataset (billing diagnoses) was not properly validated for accuracy.

Xu *et al*. (Xu et al, 2011) used a two-step process for case detection, which included document-level detection strategy of concepts related to colorectal cancer (CRC) and a patient-level case determination module. Using the two-step process provided more accurate case detection when compared to either method of the two-step process. Random Forest, Support Vector Machines, Logistic Regression and Ripper were ML-based methods that were used on the patient-level data to detect cases. There was no direct comparison of this study's method against the traditional phenotyping process although the authors did note that it was difficult to define explicit rules by manual review of aggregated data. ML-based approaches were employed to automatically find the useful patters to determine if a patient was a CRC case. Using this approach could simplify algorithm development.

The remaining three studies made comparisons between ML-based approaches as they were applied to a specific phenotyping activity. Amand (Anand and Downs, 2010) compared a Bayesian Network (BN) approach to reformulated BN using Noisy-OR, recursive Noisy-OR and adaptive Noisy-OR approaches. Wu et al. (Wu *et al*, 2010) compared Boosting and SVM to the application of identifying heart failure patients. Finally, Huang *et al*. (Huang et al, 2007) identified Type 2 diabetic patients using Naïve Bayes, IB1 and C4.5 classification techniques. There were no comparisons made to the physician-led traditional approach.

In the past year and a half, there have been two other notable studies published that used machine learning for phenotyping. Dingcheng *et al* developed an *Apriori* association rule-learning algorithm to phenotype type 2 diabetics (Dingcheng et al, 2013). This work is similar to the relational learning ILP method as they both take advantage of learning rules for phenotyping that are easily understood by human users. The primary difference between the approaches is that relational learning can directly learn from the tables of the EHR versus Apriori, which must learn from data conflated into a single table. The authors reported positive predictive values greater than 90% for their algorithm.

Carroll *et al* conducted another study using rheumatoid arthritis as the phenotype. This was a multi-site study and combined structured query phenotyping with NLP methods. The cross-site accuracy estimates were greater than 95% positive predictive value once the algorithms were adjusted for site variation. This study used NLP that was based on a ML framework and not consistent with the literature review methods stated above. The study was however noteworthy because of its multi-site nature and high predictive values for the algorithm (Carroll et al, 2012).

In summary, there are only a few articles, which present research surrounding the application of ML, using coded data, for the phenotyping process. From these evaluations, there is some evidence that suggest ML improves the accuracy of the phenotyping process. These studies applied a variety of classical or rule-based ML approaches that took advantage of data placed into a fixed length feature table for analysis. The feature tables, which are critical for the supervised learning task, were based on input from experts (physicians) and/or available

validated classified subjects. There have been no studies that have used the relational learning methods, which take advantage of the EHR's relational structure.

CHAPTER 3

Importance of Multi-modal Approaches to Effectively Identify Cataract Cases from Electronic Health Records

As noted in the two preceding chapters, there is increasing interest in using electronic health records (EHRs) to identify subjects for genomic association studies, due in part to the availability of large amounts of clinical data and the expected cost efficiencies of subject identification. In this chapter I describe the construction and validation of an EHR-based algorithm to identify subjects with age-related cataracts. The approach used in this chapter is a multi-modal strategy utilizing many of the computational methods surveyed in the previous chapter, ranging from structured database querying, natural language processing (NLP) on free-text documents and optical character recognition (OCR) on scanned clinical images. The goals are to identify cataract subjects and related cataract attributes. Extensive validation on 3657 subjects compared the multi-modal results to manual chart review. The algorithm was also implemented at participating electronic MEdical Record GEnomics (eMERGE) institutions. I demonstrate that this multi-modal computational strategy makes it possible to more efficiently and with a high degree of accuracy characterize research subjects using EHR data, thus supporting my thesis statement.

3.1 BACKGROUND

Marshfield Clinic is one of five institutions participating in the electronic MEdical Records and GEnomics (eMERGE) (eMERGE, 2010); McCarty CA *et al*, 2011; Kho *et al,* 2011). One of the goals of eMERGE is to demonstrate the viability of using electronic health

record (EHR) systems as a resource for selecting subjects for genome-wide association studies (GWAS). Marshfield's GWAS focused on revealing combinations of genetic markers that predispose subjects to the development of age-related cataracts. Cataract subtypes and severity are also important attributes to consider, and possibly bear different genetic signatures (McCarty et al, 2003). Often, clinically relevant information on conditions such as cataracts is buried within clinical notes or in scanned, hand-written documents created during office visits, making this information difficult to extract.

Cataracts are the leading cause of blindness in the world (Thylefors et al, 1994), the leading cause of vision loss in the United States (U.S.) (Congdon et al, 2004), and account for approximately 60% of Medicare costs related to vision (Ellwein & Urato, 2002). Prevalence estimates indicate that 17.2% of Americans residing in the U.S. aged 40-years and older have a cataract in at least one eye, and 5.1% have a pseudophakia/aphakia (previous cataract surgery) (Congdon et al, 2004). Age is the primary risk factor for cataracts. With increasing life expectancy, the number of cataract cases and cataract surgeries is expected to increase dramatically unless primary prevention strategies can be developed and successfully implemented.

There is a growing interest in utilizing the EHR to identify clinical populations for GWAS (1; Manolio, 2009; Wojczynski & Tiwari, 2008) and pharmacogenomics research (McCarty et al, 2011; Wilke et al, 2011; McCarty, 2010). This interest results, in part, from the availability of extensive clinical data found within the EHR and the expected cost efficiencies that can result when using computing technology. As in all research that attempts to identify and quantify relationships between exposures and outcomes, rigorous characterization of study subjects is essential and often challenging (Bickeboller et al, 2003; Schulz et al, 2004).

In this chapter, I describe the construction and validation of a novel algorithm that utilizes several techniques and heuristics to identify subjects with age-related cataracts and the associated cataract attributes using only information available in the EHR. I also describe a multi-modal phenotyping strategy that combines conventional data mining with natural language processing (NLP) and optical character recognition (OCR) to increase the detection of subjects with cataract subtypes and optimize the phenotyping algorithm accuracy for case detection. The use of NLP and OCR methods was influenced by previous work in the domain of biomedical informatics that has shown great success in pulling concepts and information from textual and image documents (Govindaraju, 2005; Milewski & Govindaraju, 2004; Piasecki & Broda, 2007). I was also able to quantify the accuracy and recall of the multi-modal phenotyping components. Finally, this algorithm was implemented at three other eMERGE institutions, thereby validating the transportability and generalizability of the algorithm. The fact that other institutions were able to run the algorithm and obtain high precision (between 95-100%), is worth noting.

3.2 SIGNIFICANCE AND CONTRIBUTION

EHR-based phenotyping is a process that uses computerized analysis to identify subjects with particular traits as captured in an EHR. This process provides the efficiency of utilizing existing clinical data but also introduces obstacles, since those data were collected primarily for patient care rather than research purposes. Previously described EHR data issues include a lack of standardized data entered by clinicians, inadequate capture of absence of disease, and wide variability among patients with respect to data availability (this availability itself may be related to the patient's health status) (Wojczynski & Tiwari, 2008; Gurwitz et al, 2010). Careful phenotyping is critical to the validity of subsequent genomic analyses (Bickeboller et al, 2003),

and a source of great challenge due to the variety of phenotyping options and approaches that can be employed with the same data (Schulze et al, 2004).

Previous investigators have demonstrated successful use of billing codes and NLP for biomedical research (Denny et al, 2010; Peissig et al, 2006; Ritchie et al, 2010; Kullo et al, 2010; Savova et al, 2010). Most often, the focus in the informatics domain is on the application and evaluation of one specific technique in the context of a disease or domain, with a goal of establishing that technique's utility and performance. For example, Savova *et al*. (Savova *et al*, 2010;) evaluated the performance of Clinical Text Analysis and Knowledge Extraction System (cTAKES) for the discovery of peripheral arterial disease cases from radiology notes. Peissig *et al*.(Peissig et al, 2006) evaluated the results of FreePharma® (Language & Computing, Inc., http://www.landc.be) for the construction of atorvastatin dose-response.

Existing research has also demonstrated the ability to use multiple techniques as part of the implementation of a phenotyping algorithm (Kullo et al, 2010), but few have attempted to quantify the benefits of a multi-modal approach (conventional data mining, NLP and OCR). Those that have were able to demonstrate the benefits of two approaches (commonly coded data in conjunction with NLP) over a single approach that was limited to a single domain (Kullo et al, 2010; Rasmussen et al, 2011). Although the use of multiple modes for phenotyping is practical, no known work has explored beyond a bimodal approach. The research presented here demonstrates the ability to implement a tri-modal phenotyping algorithm including quantification of the performance of the algorithm as additional modes are implemented.

3.3 METHODS

3.3.1 Marshfield's Study Population

The Personalized Medicine Research Project (PMRP) (McCarty et al, 2005; McCarty et al, 2008), sponsored by Marshfield Clinic, is one of the largest population-based biobanks in the U.S. The PMRP cohort consists of approximately 20,000 consented individuals who provided DNA, plasma, and serum samples along with access to health information from the EHR and questionnaire data relating to health habits, diet, activity, environment, and family history of disease. Participants in this cohort generally receive most, if not all, of their primary, secondary, and tertiary care from the Marshfield Clinic system, which provides health services throughout Central and Northern Wisconsin. This research was approved by the Marshfield Clinic's Institutional Review Board.

3.3.2 Electronic Medical Record

Founded in 1916, Marshfield Clinic is one of the largest comprehensive medical systems in the nation. CattailsMD, an internally developed EHR at Marshfield Clinic, is the primary source of EHR data for this investigation. The EHR is deployed on wireless tablets and personal computers to over 13,000 users, including over 800+ primary and specialty care physicians in both inpatient and outpatient healthcare settings. Medical events including diagnoses, procedures, medications, clinical notes, radiology, laboratory, and clinical observations are captured for patients within this system. EHR-coded data are transferred daily to Marshfield Clinic's Data Warehouse (DW) and integrated with longitudinal patient data, currently providing a median of 23 years of diagnosis history for PMRP participants. In addition to the coded data, Marshfield has over 66 million electronic clinical narrative documents, notes, and images that are available back to

1988, with supporting paper clinical charts available back to 1916. Manual review of the electronic records (and clinical charts as needed) was used as the "gold standard" when validating the EHR-based algorithms.

3.3.3 Cataract Code-based Phenotyping

Cataract "cases" were identified using an electronic algorithm that interrogated the EHR-coded data found within the DW (figure 3.1). A goal of the electronic algorithm development was to increase the number of subjects identified for the study (sensitivity), while maintaining a positive predictive value (PPV) of 95% or greater. PPV is defined as the number of accurately classified cases over the total number of cases. Cases had to have at least one cataract Current Procedural Terminology (CPT®) surgery code or multiple International Classification of Diseases (ICD-9-CM) cataract diagnostic codes. In cases where only one cataract diagnostic code existed for a subject, NLP and/or OCR were used to corroborate the diagnosis. Cataract "controls" had to have an optical exam in the previous 5 years with no evidence of cataract surgery or a cataract diagnostic code or indication of a cataract when using either NLP and/or OCR. Since the focus of the eMERGE study was limited to age-related cataracts, subjects were excluded if they had any diagnostic code for congenital, traumatic, or juvenile cataract. Cases were further restricted to be at least 50-years-old at the time of either cataract surgery or first cataract diagnosis, and controls had to be at least 50-years-old at their most recent optical exam.

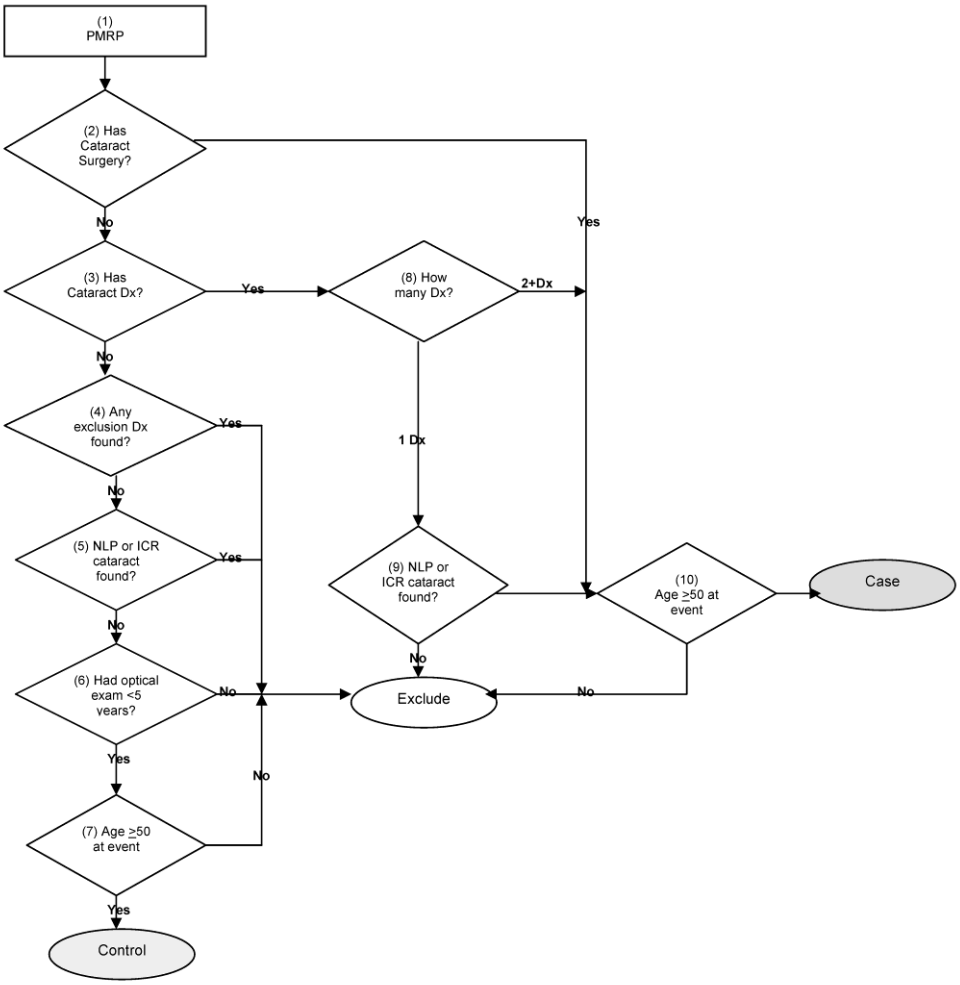


Figure 3.1: eMERGE Cataract phenotyping algorithm. Overview of the cataract algorithm logic used when selecting the cataract cases and controls for the electronic MEdical Record and GEnomics (eMERGE) genome-wide association study. Cataract cases were selected if the subject had either a cataract surgery, or 2+ cataract diagnoses, or 1 cataract diagnosis with either an indication found using Natural Language Processing (NLP) or Optical Character Recognition (OCR). Controls had to have an optical exam within 5 years with no evidence of a cataract. Both cataract cases and controls had to be age 50 or older with controls requiring the absence of the exclusion criteria. The details of this algorithm are published on the eMERGE website (eMERGE, 2010).

3.3.4 Cataract Subtype Multi-modal Phenotyping

A multi-modal phenotyping strategy was applied to the EHR data and documents to identify information pertaining to nuclear sclerotic, posterior sub-capsular, and cortical (N-P-C) cataract subtypes, severity (numeric grading scale), and eye. Over 3.5 million documents for the PMRP cohort were pre-processed using a pattern search mechanism for the term “cataract”. The strategy (figure 3.2) consisted of three methods to identify additional cataract attributes:

conventional data mining using coded data found in the DW, NLP used on electronic text documents, and OCR used on scanned image documents. Conventional data mining was used to identify all subjects having documented N-P-C subtype (ICD9 codes 366.14–366.16).

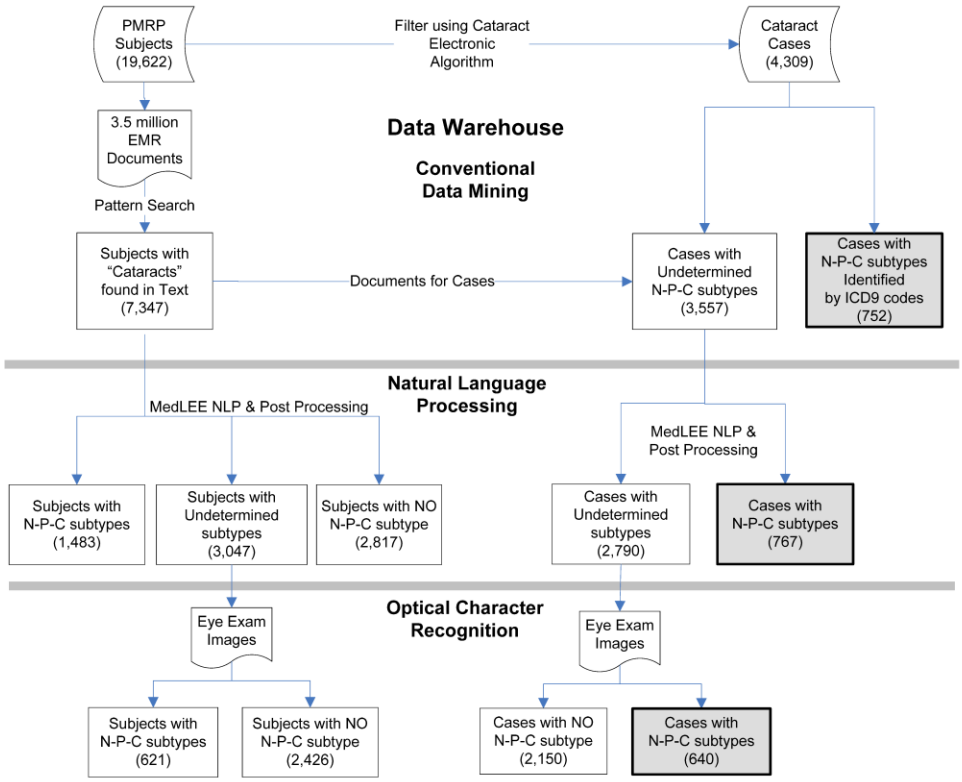


Figure 3.2: Multi-modal Cataract Subtype Processing. Overview of the information extraction strategy used in multi-modal phenotyping to identify nuclear sclerotic, posterior sub-capsular and/or cortical (N-P-C) cataract subtypes. This figure depicts the N-P-C subtype yield using two populations: 1) the left-most path of the figure denotes unique subject counts for the entire Personalized Medicine Research Project cohort; 2) the right-most path denotes unique subject counts for the identified cataract cases. A hierarchical extraction approach was used to identify the N-P-C subtypes. If a subject had a cataract subtype identified by an ICD-9 code, Natural Language Processing (NLP) or Optical Character Recognition (OCR) was not utilized. Cataract subtypes identified using NLP had no subsequent OCR processing.

Prior to using NLP to identify N-P-C subtypes, a domain expert was consulted regarding documentation practices surrounding cataracts, who determined that the term "cataract" should always appear within a document for it to be considered relevant to the N-P-C subtypes. The reasoning behind this was to avoid any potential ambiguity when terms related to cataract subtype (i.e., "NS" as an abbreviation for “nuclear sclerotic”) appeared in a document with no

further support related to a cataract. As all clinical documents were of interest, not just ones from ophthalmology, this rule enabled the application of a filter to the documents to be processed by NLP. The Medical Language Extraction and Encoding (MedLEE) NLP engine (MedLEE, 2011), developed by Friedman and colleagues (Friedman et al, 2004) at Columbia University, was tuned to the ophthalmology domain for this specific task to process documents from PMRP patients. MedLEE was chosen for its demonstrated performance in other studies (Melton & Hripcsak, 2005; Friedman et al, 2004) and also given the experience of one of the authors (JS) with MedLEE in previous studies (Friedman et al, 1995; Starren & Johnson, 1996; Starren et al, 1995 (1 & 2)). While MedLEE was chosen for Marshfield's implementation, the multi-modal approach was created to be NLP engine-neutral, and other sites utilized the cTAKES (Savova et al, 2010) engine with comparable results.

The NLP engine tuning involved iterative changes to the underlying lexicon and rules based on a training corpus of 100 documents. MedLEE parses narrative text documents and outputs eXtensible Markup Language (XML) documents, which associate clinical concepts with Unified Medical Language System (UMLS) Concept Unique Identifiers (CUIs (National Library of Medicine, 2003; Lindberg, Humphreys and McCray, 1993) with relevant status indicators, such as negation status. To identify general cataract concepts and specific cataract subtypes, specific CUIs were queried based on MedLEE's output. Additional CUIs were used to determine in which eye the cataract was found. A regular expression pattern search was performed on MedLEE attributes to identify severity of the cataract and certainty of the information provided. Refer to figure 3.3 for an overview of the NLP process.

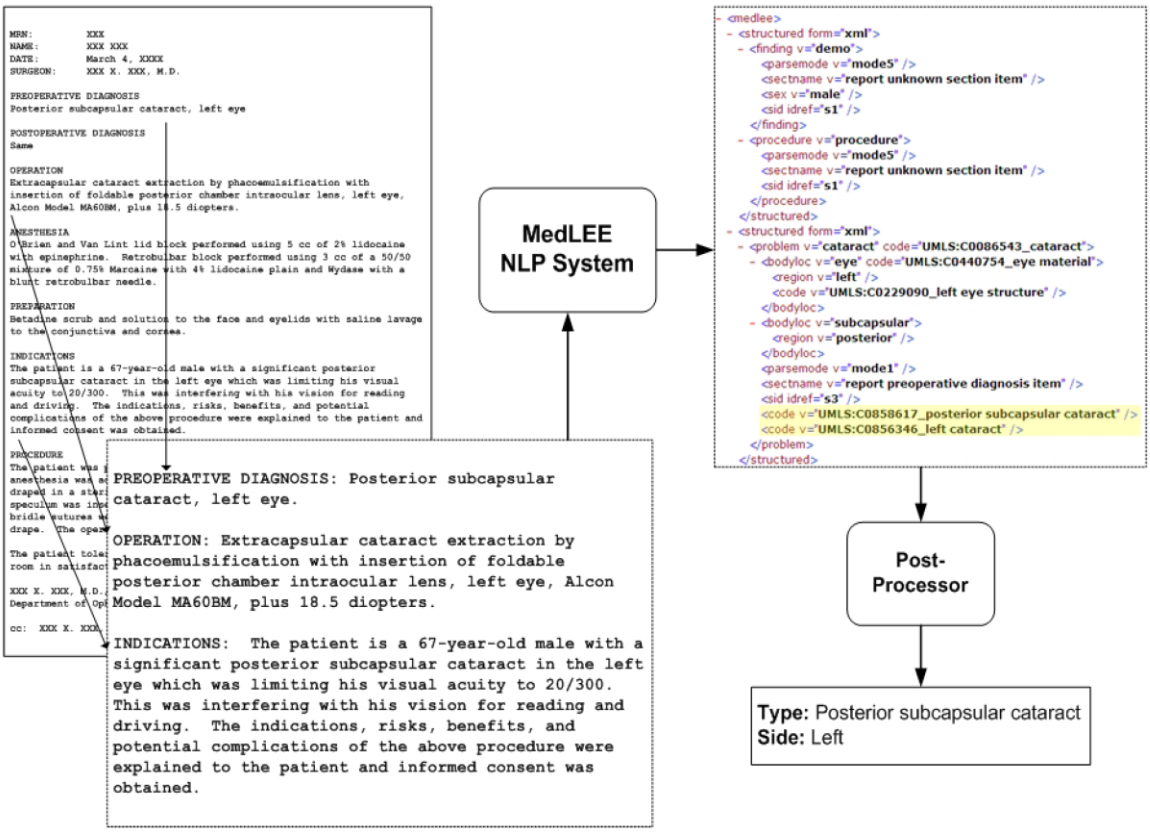


Figure 3.3: Natural Language Processing of Clinical Narratives. Textual documents containing at least one reference of a cataract term were fed into the MedLEE Natural Language Processing (NLP) engine and then tagged with appropriate UMLS Concept Unique Identifiers (CUIs) before being written to an XML formatted file. Post-processing consisted of identifying the relevant UMLS cataract CUIs and then writing them along with other patient and event identifying data to a file that was used in the phenotyping process.

For subjects with no cataract subtype coded or detected through NLP processing, ophthalmology image documents were processed using an OCR pipeline developed by the study team. The software loaded images and encapsulated the Tesseract (Tesseract-OCR, 2010) and LEADTOOLS (LEADTOOLS® , 2010) OCR engines into a single pipeline that processed either digital or scanned images for detecting cataract attributes (figure 3.4). From each image an identification number and region of interest were extracted and stored in a Tagged Image File Format (TIFF) image. The TIFF image was then passed through both OCR engines with the results being recorded independently. Neither the Tesseract nor LEADTOOLS engine was 100%

accurate and often both engines misclassified characters in the same way (i.e., “t” instead of “+”). Regular expression processing was used to correct these misclassifications. The output from this process yielded a new string that was matched against a list of acceptable results. If after two iterations the string could not be matched, it was discarded. Otherwise, the result was loaded into a database, similar to the NLP results. Details of this process are the subject of another paper (Rasmussen et al, 2011).

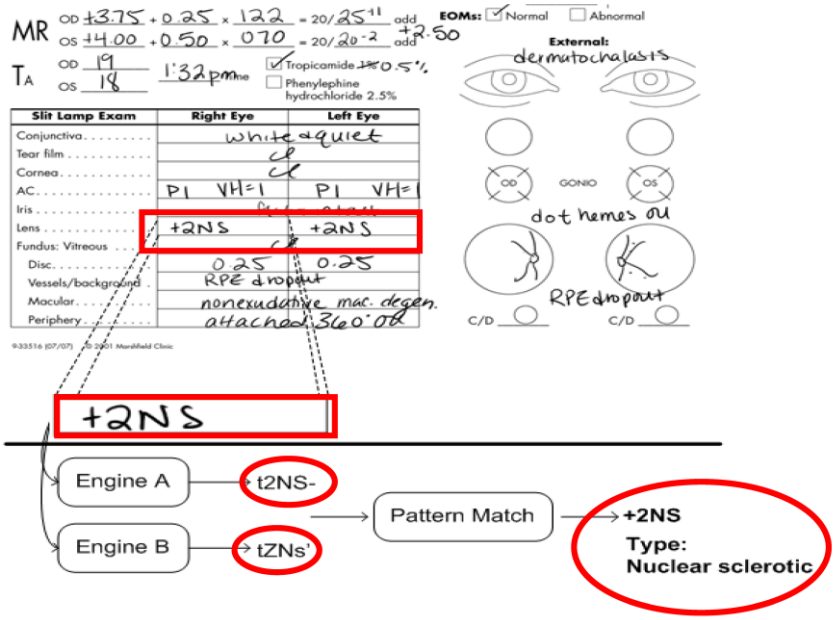


Figure 3.4: Optical character recognition utilizing electronic eye exam images. Image documents were processed using the LEADTOOLS and Tesseract Optical Character Recognition (OCR) engines. A tagged image file format (TIFF) image was pasted through the engines with results being recorded independently. Common misclassifications were corrected using regular expressions, and final determinations were made regarding subtype and severity.

3.3.5 Validation

All electronically identified potential surgical cataract cases were manually reviewed by a trained research coordinator who verified the presence or absence of a cataract by using the medical chart and supporting documentation. A board-certified ophthalmologist resolved questions surrounding classifications. In addition, stratified random samples were selected to

validate potential cases with diagnoses but no surgery, potential controls, and each cataract attribute for the multi-modal approach.

3.3.6 Analysis

Validation results for the algorithms are summarized with standard statistical information retrieval metrics, sensitivity (or recall), specificity, positive predictive value (PPV) (or precision) and negative predictive value (NPV) (Altman & Bland, 1994; Altman &Bland, 1994 (2)). Sampling for the validation was stratified with different sampling fractions in different strata (e.g., 100% of surgeries were reviewed). Accuracy and related statistics have been weighted based on the sampling fractions to reflect the test properties as estimated for the full PMRP cohort. Final cataract algorithm and overall cataract subtype 95% confidence interval statistics in tables 3.1 and 3.2 are based on 1000 bootstrap samples (Efron & Tibshrani, 1003; Zhou et al, 2002).

3.3.7 Cross-Site Algorithm Implementation

Group Health Research Institute/University of Washington (GH), Northwestern University (NU), and Vanderbilt University (VU) belong to the eMERGE consortium and are performing genome-wide association studies (GWAS) using EHR-based phenotyping algorithms to identify patients for research. The GWAS activities conducted by each institution had approval from their respective internal review boards. Group Health conducted GWAS on dementia; Northwestern University utilized Type 2 diabetics for their GWAS; and Vanderbilt University conducted GWAS on determinants of normal cardiac conduction (Denny et al, 2010). EHRs used by these institutions included an in-house developed EHR (Vanderbilt University) and the vendors Epic and Cerner (Group Health and Northwestern University). The cataract

algorithm (figure 3.4) was implemented at three additional sites. Due to lack of use of digital forms at the other institutions, only conventional data mining using billing codes and NLP portions of the algorithm were implemented, with adjustments to some of the diagnostic and procedural codes representing inclusion or exclusion criteria.

3.4 RESULTS

The PMRP cohort consisted of 19,622 adult subjects as described previously (Waudby et al, 2011). The number of years of available EHR data for subjects within the PMRP cohort ranged from <1 year to 46 years, with a median of 23 years. Interrogating the coded DW identified 4835 subjects that had either a cataract surgery and/or a cataract diagnosis code. Of the 1800 subjects that had a cataract surgery code, 1790 subjects (99.4%), had a corresponding cataract diagnosis code recorded in the EHR.

Cases and controls were manually validated to obtain cataract status and clinical attributes on 3657 subjects during the course of this study. The cataract phenotype validation results are shown in table 3.1. The final algorithm developed to identify cataract cases and controls is shown in figure 3.1, with the full algorithm publically available at http://gwas.org (emerge website). Cases were identified for eMERGE using the "Combined Definition" from table 3.1. Controls had to meet the definition previously provided. This algorithm yielded 4309 cases with a PPV of 95.6% and 1326 controls with a NPV of 95.1%.

The multi-modal phenotyping strategy was used on all PMRP subject data to identify cataract subtype attributes (figure 3.1). There were 7347 out of the 19,622 subjects (37%), who had a cataract reference that triggered further NLP or OCR processing to identify the N-P-C subtypes. Of the 4309 subjects identified as cataract cases (figure 3.2), 2159 subjects (50.1%), had N-P-C subtypes identified using the multi-mode methods. Of these, conventional data

mining identified 752 unique subjects (34.8%); NLP recognized an additional 767 distinct subjects (35.5%), and OCR further added 640 unique subjects (29.6%) with N-P-C subtypes. Table 3.2 reports summary validation results for each of the N-P-C cataract subtypes with respect to the 4309 cases that were identified using the combined cataract algorithm (table 3.1). I present statistics that are weighted to the full PMRP cohort based on the sampling fractions and statistics that are based on the available relevant data. For each group, all approaches (conventional data mining, NLP, OCR) had high predictive values meeting the threshold guidelines. Combining all multi-modal strategies produced promising results and met the threshold for quality. Appendix 1 presents detailed validation statistics for all cataract subtypes by all multimodal methods for the weighted and relevant data sampling fractions.

Table 3.1: Cataract phenotype validation results.

Electronic Approaches	Cohort Size	Manually Reviewed	Electronic Case Yield	Manual Case Yield	True Positives	PPV	NPV	Sensitivity	Specificity
Surgery	19,622	3657	1800	3270	1747	100%	82.8%	36.9%	100%
1+Dx (no surgery)	17,822	1910	3035	1523	1507	86.6%	97.0%	85.6%	97.2%
2+Dx (no surgery)	17,822	1910	2047	1523	1046	92.1%	92.5%	61.4%	98.9%
3+Dx (no surgery)	17,822	1910	1592	1523	842	94.4%	90.3%	44.8%	99.4%
NLP and/or OCR +1 Dx (no surgery)	1470	350	462	286	272	93.4%	76.8%	64.8%	96.2%
Combined Definition[1]	**19,622**	**3657**	**4309**	**3270**	**3065**	**95.6%**	**95.1%**	**84.6%**	**98.7%**
(95% CI)[2]						(94.7, 96.4)	(92.6, 96.9)	78.3, 89.7)	98.5, 99.0)

[1]Combined electronic phenotype definition: surgery, and/or Inclusion Dx 2+, and/or (Inclusion Dx1 + NLP/OCR)

[2]95% Confidence interval (CI) based on 1000 bootstrap sample.

Validation results for the cataract phenotyping effort are based on unique subject counts. Results are presented in a step-wise fashion starting with Surgery. The surgery cases were removed prior to presenting the diagnosis (Dx) results. The combined electronic phenotype definition consists of subjects having a cataract surgery or 2+ Dx or 1 Dx with either a confirmed recognition from natural language processing (NLP) or optical character recognition (OCR).

Table 3.2: Detailed results for the cataract subtype multi-modal validation.

Subtype	Relevant Cohort Size[4]	Manually Reviewed[4]	Electronic Subtype Yield[5]	Manual Subtype Yield[5]	True Positives	Electronic Approach Weighted Statistics based on Sampling Strata				Manual Approach Statistics based on Relevant Data			
						PPV[6]	NPV[6]	Sensitivity[6]	Specificity[6]			Sensitivity	Specificity
Nuclear Sclerotic[1]													
CDM on Coded Diagnosis	752	629	493	612	410	99.5%	1.9%	13.8%	96.2%	99.5%	3.8%	67.0%	80.0%
Overall (CDM/NLP/OCR)[2] (95% CI)[3]	**4236**	**3118**	**1849**	**2946**	**1654**	**99.9**% (99.7, 100)	**3.7**% (2.7, 4.7)	**55.8**% (53.9, 57.6)	**96.2**% (9.03, 100)	**99.9**% (99.7, 100)	**3.7**% (2.6, 4.7)	**56.1**% (54.2, 58.0)	**96.1**% (90.0, 100)
Posterior Sub-capsular[1]													
CDM on Coded Diagnosis	752	629	287	287	224	97.0%	66.5%	21.5%	99.6%	97.0%	81.5%	78.0%	97.5%
Overall (CDM/NLP/OCR)[2] (95% CI)[3]	**4236**	**3118**	**529**	**1036**	**425**	**95.1**% (92.9, 97.1)	**72.3**% (70.6, 74.2)	**40.9**% (37.7, 43.8)	**98.6**% (98.0, 99.2)	**95.1**% (93.0, 96.8)	**72.3**% (70.4, 74.0)	**41.0**% 38.0, 43.9)	**98.6**% (98.0, 99.1)
Cortical[1]													
CDM on Coded Diagnosis	752	630	3	384	1	100.0%	25.1%	0.0%	100.0%	100.0%	32.3%	0.3%	100.0%
Overall (coded/NLP/OCR)[2] (95% CI)[3]	**4236**	**3119**	**765**	**2102**	**674**	**95.7**% (94.1, 97.2)	**32.0**% (29.9, 34.1)	**31.9**% (30.0, 33.8)	**95.8**% (94.3, 97.2)	**95.7**% (94.3, 97.3)	**32.1**% (30.0, 34.01)	**32.1**% (29.9, 34.1)	**95.7**% (94.2, 97.2)

PPV= positive predictive value ; NPV= negative predictive value; CDM= conventional data mining; NLP= natural language processing; OCR= optical character recognition

[1]On 4309 cases meeting cataract phenotype definition.

[2]Weighted average using CDM, NLP, and OCR strategies to reflect the properties as would be expected for the full PMRP cohort.

[3]95% Confidence Interval (CI) based on 1000 bootstrap samples.

[4]Unique case counts

[5]Yield represents total number of cataract subtyped identified. There may be up to 2 subtypes noted for each subject, but the subject will only be counted once.

[6]Statistics are weighted based on sampling fractions to reflect the properties as would be expected for the full PMRP cohort.

Shown are the multi-modal cataract subtype validation statistics. Statistics were calculated using two different approaches: 1) weighted based on the sampling strata, to reflect the properties as would be expected for the full PMRP cohort; and 2) based on the availability of relevant data. A multi-mode cataract sub-typing strategy, consisting of conventional database mining (CDM), natural language processing (NLP) on electronic clinical narratives and optical character recognition (OCR) utilizing eye exam images was used to electronically classify cataract subtypes. The electronic subtype classifications were then compared to manually abstracted subtype classifications to determine the accuracy of the multi-modal components.

Table 3.3: Severity and location validation results.

	Relevant Cohort Size	Manually Reviewed	Electronic Attribute Yield	Manual Attribute Yield	True Positives	PPV	NPV	Sensitivity	Specificity
Severity 2+ Detected[1,2]									
NLP	3862	3144	198	2510	189	98.6%	27.1%	6.9%	99.7%
OCR (no NLP)	3236	2952	180	2321	150	91.4%	29.3%	6.9%	98.4%
Location Detected[1]									
NLP	3862	2502	2265	2422	1529	99.7%	7.7%	63.1%	93.8%
OCR (no NLP)	3236	968	21	892	10	100.1%	7.8%	1.1%	100.0%

PPV= positive predictive value ; NPV= negative predictive value; NLP= natural language processing; OCR= optical character recognition

[1]On 4309 cases meeting cataract phenotype definition.

[2]If cataracts found via abstraction and severity unknown, then severity (per abstraction) set to 'No.'

Shown are the cataract severity and location (left or right eye) validation results for the 4,309 cataract subjects who were classified as cases using the electronic cataract phenotyping definition. Natural language processing (NLP) was utilized first on electronic clinical narratives to identify cataract severity and/or location. Eye exam document images were processed using optical character recognition (OCR) if the severity and location could not be determined using NLP. There were no coded severities or location data found within the electronic health record.

Figure 3.5 illustrates the overlap of data sources for classifying cases with the nuclear sclerotic (NS) cataract subtype which is by far the most common. The NLP approach identified the majority, 1213 (65.6%) subjects with the subtype. OCR methods identified 813 (44%) subjects, and conventional data mining identified 493 (26.7%) subjects.

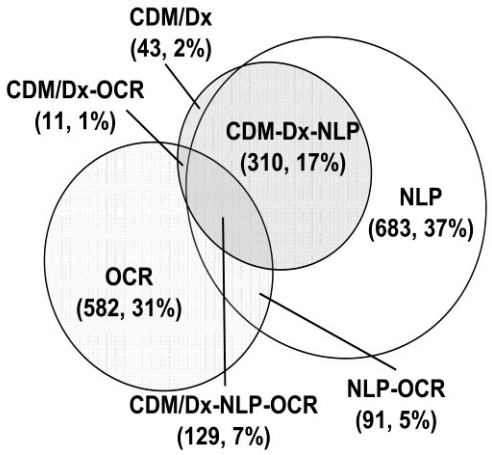


Figure 3.5: Nuclear sclerotic cataract subtype multi-modal approaches. Illustrates the overlap between multi-modal phenotyping approaches when phenotyping for nuclear sclerotic (NS) cataract subtypes. The largest subtype yield comes from natural language processing (NLP) with NS being identified for 1213 unique subjects. This is followed by the optical character recognition (OCR) approach, which identified 813 unique subjects. Conventional data mining (CDM/Dx) using diagnostic

Figure 3.6 shows the contribution of multi-modal phenotyping to increase the number of subjects with NS subtypes from 493 (11.6%) to 1849 (43.6%). In actual practice, the conventional data mining approach would be used first to identify subtypes because it requires the least effort. The remaining documents would then be processed using NLP to identify more subtypes, and if there were subjects remaining, the OCR approach would be used to identify subtypes. In this study, OCR required the most effort to setup and train the OCR engine to gain subtype information. All of the methods had a high PPV, which was required.

Validation results for cataract severity and location are shown in table 3.3. There were no coded values for either cataract severity or location found in the EHR, so the only way to obtain that information was to rely solely on NLP or OCR. NLP identified cataract location (left or right

eye) with high reliability in 53% of the subjects. The OCR yield presented in table 3.3 is low because only the remaining subjects that did not have NLP-identified locations were processed.

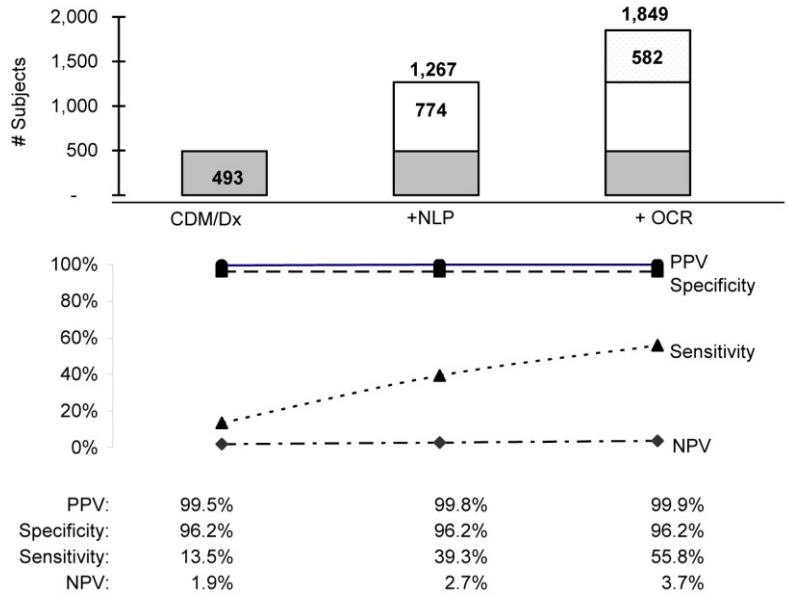


Figure 3.6: Multi-modal yield and accuracy for nuclear sclerotic cataract subtype. Illustrates a step-wise approach that was used to identify subjects with nuclear sclerotic (NS) subtypes. Conventional data mining (CDM/Dx) using ICD9 diagnostic codes was used first because it required the least effort. Natural language processing (NLP) was applied to the remaining subjects if a NS subtype was not identified and optical character recognition (OCR) was used if the previous approaches did not yield a subtype. Out of a possible 4309 subjects having cataracts, 1849 subjects had indication of a NS subtype. Both yield (represented by the number of unique subjects having a NS subtype) and accuracy (represented by positive predictive value (PPV), Specificity, Sensitivity and negative predictive value (NPV)) are presented for each approach. This study used PPV as the accuracy indicator. The number of unique subjects with NS subtypes increased using the multi-modal approach while maintaining a high PPV.

3.4.1 External Validation

The other eMERGE institutions were able to contribute 1386 cases and 1360 controls to Marshfield's cataract GWAS study. GH did not explicitly validate conventional data mining (CDM)-based phenotype case and control definitions by manual chart review. Instead, a separate review using NLP-assisted chart abstraction methods was conducted for 118 subjects selected from the GH cohort of 1042 (931 cases and 111 controls) who had at least one chart note

associated with a comprehensive eye exam in the EHR. All 118 (100%) were in the CDM-based eMERGE phenotype case group, and all had cataract diagnoses documented in their charts. Additionally, cataract type was recorded for 108 (92%) of the subjects. The remaining 10 (8%) did not specify cataract type at all or with insufficient detail to satisfy study criteria. Though the 118 charts reviewed for the cataract type analysis does not constitute a statistically significant validation sample, the fact that 100% were in the cataract case group is reassuring. VU sampled and validated 50 cases and 42 controls that were electronically identified against classifications determined by a physician when reviewing the clinical notes. Results from this validation indicated a positive predictive value of 96%.

3.5 DISCUSSION

The EHR represents a valuable resource for obtaining a longitudinal record of a patient's health and treatments through on-going interactions with a healthcare system. The health information obtained from those interactions can be stored in multiple formats within the EHR, which may vary between institutions and may present significant challenges when trying to retrieve and utilize the information for research (Wojczynski & Tiwari, 2008). This work demonstrates that electronic phenotyping using data derived from the EHR can be successfully deployed to characterize cataract cases and controls with a high degree of accuracy. It also demonstrates that a multi-modal phenotyping approach, which includes the use of conventional data mining, NLP, and OCR, increased the number of subjects identified with N-P-C subtypes from 752 (17.5%) to 2159 (50.1%) of the available cataract cases. Multi-modal phenotyping to

increase the number of subjects with nuclear sclerotic subtypes went from 493 to 1849 (figure 6) while maintaining a high PPV, as required.

These multi-modal methods can be applied to a variety of studies beyond the current one. In many clinical trials, the actual patients who are sought cannot be identified purely by the presence of an ICD code. As a simple example, squamous and basal cell skin cancer share the same ICD code, yet have different treatment and prognosis. Efficiently identifying patients with one or the other diagnosis requires using text documents or scanned documents. Similarly, many studies include historical inclusion or exclusion criteria. Such historical data are poorly incorporated in standard codes, but are frequently noted in textual or scanned documents. The addition of multimodal methods can improve the efficiency of both interventional and observational studies. These techniques can also be applied to quality and safety issues to identify adverse events or process failures. When used in these domains, the tools will need to be biased toward sensitivity, rather than toward specificity, as they were in this study.

A commonly repeated saying in the EHR phenotyping domain is that "The absence of evidence is not evidence of absence" (Sagen, 2013) In the case of this study, that could be due to cataract surgery at an outside institution, and presence of an early cataract that was not detected on routine examination. Therefore, in this and other phenotyping studies, it is critical to actively document the absence of the target phenotype when defining the control group. In the case of this study, intraocular lenses that were placed during cataract surgery were observed and documented on clinical eye exams. All subjects were required to have an eye exam to prevent missing cataract surgeries, regardless of where they took place.

An extensive validation comparison between the multi-modal results to manually abstracted results (considered the gold-standard) has been presented. Although the validation statistics for the final electronic algorithms met the accuracy threshold (PPV $\geq$ 95%), the potential exists to improve the accuracy by modifying the algorithm to exclude additional misclassified subjects. For example, when classifying cataract cases, the "Combined Definition" (table 3.1), was used to maximize the yield of cases at the expense of lowering the accuracy threshold to a PPV of 95.6%, when compared to using only subjects having a cataract surgery (PPV of 100% and case yield of 1800, table 3.1). The study could achieve 100% accuracy, albeit with small numbers (low sensitivity). However, excluding subjects to improve accuracy is something that would have to be discussed when conducting individual studies. Further, improvement in the statistics would have to be balanced against the possibility that the resulting cases would be less representative of the population of interest.

The portability of the cataract algorithm to other EHRs has also been demonstrated. Although EHRs differed between eMERGE institutions, basic information captured as part of patient care is similar across all of the institutions (Kho et al, 2011). An important outcome of this research is the demonstrated performance of the cataract algorithm at three institutions with separate EHR systems using only structured billing codes. The cataract algorithm was validated at three other institutions with similarly high PPVs. The ability to successfully identify N-P-C subtype information improved with availability of electronic textual and image documents.

One weakness of this study was the use of a single Subject Matter Expert (SME) for the development of NLP and OCR tools. The recording of cataract information tends to be more

uniform in the medical record than many other clinical conditions; therefore, the role of the SME was primarily to assist in identifying any nuances particular to the group of Ophthalmologists practicing at the Marshfield Clinic. Subsequent high accuracy of the NLP tool when applied to other institutions affirmed the consistency of documentation. Clinical domains with higher variability may require multiple SMEs.

A limitation of this study is that the multi-modal strategy was applied to a single clinical domain area. Although this study demonstrates the utility of this approach for cataracts, there may be other clinical domains where the multi-modal strategy may not be cost-effective or even possible. For example, if much of the domain information is in coded formats, the resource to implement a multi-mode algorithm would not justify the use of this strategy. There have been other studies that utilized a bimodal approach to phenotyping (CDM and NLP) which have shown an increase in PPV values (Denny et al, 2010; Ritchie et al, 2010; Kullo et al, 2010). I realize that the application of a multi-mode strategy should be evaluated on a study-by-study basis with consideration given to the time and effort of developing tools, the number of subjects required to conduct a meaningful study, and the sources of available data. Utilizing a multi-mode strategy for identifying information that is not often reliably coded is an excellent use of resources when compared to the alternative manual abstraction method. Benefits of using automation over manual record abstraction are reuse and the portability of the electronic algorithms to other EHRs.

A second limitation is the lack of validation of a third mode (specifically, image documents containing handwritten notes) at other eMERGE institutions that implemented this

algorithm. Results describing the benefits of applying a third mode are then localized to Marshfield Clinic. This does, however, demonstrate that algorithms developed with multiple phenotyping modes may do so in a way that modularizes each mode such that it can be removed from the algorithm if a replicating institution does not have the same information in their EHR.

A criticism of this study may be the use of a domain expert to identify features for model development in the CDM, NLP, and OCR approaches. Other studies have used logistic regression to identify attributes for CDM and NLP. Other machine learning techniques may also prove beneficial in classification tasks; however, the involvement of domain experts is the current standard in studies involving identification of research subjects or samples. The ophthalmology domain expert not only provided guidance on where to find the information, but also provided features that were representative and commonly used in Marshfield Clinic's ophthalmology practice. I believe that domain expert knowledge was critical to the success of this study, although introducing automatic classification in conjunction with a domain expert would be of interest in future research. Implementing the cataract algorithm at other eMERGE institutions required EHR domain knowledge rather than ophthalmology domain knowledge.

This study underscores a previously-stated caution regarding the use of coded clinical data for research: just because a code exists in the code-set does not guarantee that clinicians use it. Cataract subtype codes exist in ICD-9 coding, but as seen in figure 3.5, fewer than half of the NS cases identified were coded as such. From a clinical standpoint, this is not surprising, because cataract subtype has little impact on either billing or clinical management. As a result of this study, the Marshfield ophthalmology specialty is considering workflows that will emphasize

coded data capture of subtypes. Current documentation practices support ink-over-forms (figure 3.3). A hybrid approach utilizing ink-over-forms combined with coded data capture to document cataract subtypes could be developed. Although NLP and OCR are viable strategies for obtaining subtype information, utilizing conventional data mining would be most cost-effective for obtaining cataract subtypes.

3.6 CONCLUSION

The utilization of ICD9-CM codes and CPT® codes for the identification of cataract cases and controls are generally accurate. This study demonstrated that phenotyping yield improves by using multiple strategies with a slight trade-off in accuracy, when compared to results of a single strategy. Most EHRs capture similar types of data for patient care. Because of this, the portability of electronic phenotyping algorithms to other EHRs can be accomplished with minor changes to the algorithms. The decision to use a multi-mode strategy should be evaluated on an individualized study basis. Not all clinical conditions will require the multiple approaches that were applied to this study.

3.7 CONTRIBUTORSHIP TO THIS CHAPTER

Peggy Peissig prepared the initial draft of the chapter. Richard Berg carried out the statistical analyses. James Linneman developed the electronic algorithm to identify cases/controls and created the data-bases and data sets. Carol Waudby completed the data abstraction. Luke Rasmussen configured and executed the NLP and OCR programs. Peggy Peissig and Justin Starren oversaw the informatics components of the study. Lin Chen, Russell

Wilke, and Cathy McCarty were the content experts and provided training for data abstraction. Cathy McCarty was Principal Investigator and is responsible for the conception, design, and analysis plan. Joshua Denny, Jyotishman Pathak, David Carrell, and Abel Kho oversaw informatics activities at other eMERGE institutions. All authors read and approved the final manuscript.

This chapter was published as original research in the Journal of American Informatics Medical Association in 2012 (Peissig et al, 2012).

CHAPTER 4

Identifying Adverse Drug Events by Relational Learning

In the previous chapter we learned that using multiple computational methods for extracting information out of the electronic health record increased the number of subjects identified for research while maintaining a high level of accuracy. This chapter presents another example of how an innovative computational method can be used to identify patients, in this case for adverse drug event (ADE) surveillance using the EHR, thus providing efficiencies.

The pharmaceutical industry, consumer protection groups, users of medications and government oversight agencies are all strongly interested in identifying adverse reactions to drugs. While a clinical trial of a drug may use only a thousand patients, once a drug is released on the market it may be taken by millions of patients. As a result, in many cases ADEs are observed in the broader population that was not identified during clinical trials. Therefore, there is a need for continued, post-marketing surveillance of drugs to identify previously unanticipated ADEs. While the goal of post-marketing surveillance on the surface seems different from phenotyping, I claim that in a more abstract view it is an analogous task: I believe there is (or may be) a shared property, or ADE phenotype, among patients on the drug, and tried to derive a definition of this property or group of patients from the EHR data. This chapter casts this problem as a machine learning task, related to *relational subgroup discovery*, and provides an initial evaluation of this approach based on experiments with an actual EHR and known adverse drug events.

4.1 BACKGROUND

Adverse drug events (ADEs) are estimated to account for 10-30% of hospital admissions, with costs in the United States alone between 30 and 150 billion dollars annually (Lazarou, Pomeranz, and Corey 1998), with more than 180,000 life threatening or fatal ADEs annually, of which 50% could have been prevented (Gurwitz et al. 2003). Although the U.S. Food and Drug administration (FDA) and its counterparts elsewhere have preapproval processes for drugs that are rigorous and involve controlled clinical trials, such processes cannot possibly uncover everything about a drug. While a clinical trial of a drug might use only a thousand patients, once a drug is released on the market it may be taken by millions of patients. As a result, additional information about possible risks of use is often gained after a drug is released on the market to a larger, more diverse population.

This chapter proposes machine learning as a post-marketing surveillance tool in order to predict and/or detect adverse reactions to drugs from electronic health records (EHRs, also known as electronic medical records, or EMRs). I apply this approach to actual EHR datasets. This task poses several novel challenges to the Machine Learning (ML) community. Although some of these challenges have been noted in previous chapters, they warrant repeating.

1. One cannot assume that we know in advance what adverse event (ADE), a particular drug might cause. In some cases, we may suspect a specific ADE, such as increased risk of heart attack (myocardial infarction, or MI); in such a case, supervised learning can be employed with MI as the class variable. But if we do not know the ADE in advance, what class variable can we use? I propose using the *drug* itself as the class variable and claim

that, while I already know who is taking the drug, examination of a model that accurately predicts those subjects, will inform us and predict the actual entity of interest (the ADE).

2. The data are *multi-relational*. Several objects such as doctors, patients, drugs, diseases, and labs are connected through relations such as visits, prescriptions, diagnoses, etc. If traditional ML techniques are to be employed, they require flattening the data into a single table. All known flattening techniques, such as computing a join or summary features result in either (1) changes in frequencies on which machine learning algorithms critically depend or (2) loss of information.
3. There are *arbitrary* numbers of patient visits, diagnoses and prescriptions for different patients, i.e., there is no fixed pattern in the diagnoses and prescriptions of the patients. It is incorrect to assume that there are a fixed number of diagnoses or that only the last diagnosis is relevant. To predict ADEs for a drug, it is important to consider the other drugs prescribed for the patient, as well as past diagnoses, procedures, and laboratory results.
4. Since all the preceding events and their interactions are temporal, it is important to explicitly model time. For example, some drugs taken at the same time can lead to side-effects, while in other cases one drug taken after another can cause a side-effect. As I demonstrate in these experiments, it is important to capture such interactions to be able to make useful predictions.
5. There are lessons from pharmacoepidemiology about how to construct cases and controls – positive and negative examples – as well as how to address confounders. Otherwise my

methods will simply identify disease conditions associated with the drug for other reasons, such as drug indications or conditions correlated with use of the drug for other reasons.

Based on the preceding motivation, this chapter presents a machine learning approach studying an important real world, high-impact task—identifying ADEs—for which the Marshfield Clinic EHR data is available. The chapter shows how relational learning (Lavrac and Dzeroski 1994; De Raedt, 2008) is especially well-suited to the task, because of the multi-relational nature of EHR data. In addition to the importance of the ADE application, this chapter also provides the following technical lessons for ML that should be applicable to a number of other domains as well.

1. In some ML applications, we may not have observations for the class variable. For example, we might hypothesize an unknown genetic factor in a disease or an unknown subtype of a disease. In such situations, we typically resort to unsupervised learning. The task of identifying previously unanticipated ADEs is such a situation – without a hypothesized ADE, how can we run a supervised learning algorithm to model it? Without knowing in advance that MI is an ADE for Cox2ibs, how can we provide supervision such that the algorithm will predict that MI risk is raised by these drugs? The problem can be addressed by running supervised learning in a manner that might seem to be "in reverse" to learn a model to predict who is on a Cox2ib. If we can identify some subgroup of Cox2ib patients based on the events occurring after they start Cox2ib, this can provide evidence that the subgroup might be sharing some common effects of

Cox2ib. I anticipate this same approach can also be applied to other situations where the class variable of interest is not observed.

2. Useful ideas from epidemiology were considered for us in our analysis. For example, treating each patient as his/her own control, or by drawing as positive examples patients and their data after they begin use of a drug and as negative examples the same patients but before they begin use of the drug is an option for defining cases and controls. Another idea obtained from epidemiology is to use a domain-specific scoring function that includes normalization based on other drugs and other conditions. I introduce to epidemiology the notion of learning rules to characterize ADEs, rather than simply scoring drug-condition pairs, which require the ADE to correspond to an already defined condition.
3. Finally, this chapter reinforces the need for iteration between human and computer in order to obtain the models that provide the most insight for the task. In ADE identification, rules that are predictive of drug use can be taken as candidate ADEs, but these candidate ADEs must then be vetted by a human expert. If some of the rules are found to still capture other factors besides drug effects such as indications, then these rules should be discarded. I refer to this lesson as Iterative Interaction. Note that the prediction is reverse not only in terms of causality, but more importantly in terms of the label of interest.

4.2 MACHINE LEARNING FOR PREDICTING ADES

Learning adverse events can be defined as follows:

Given: Patient data (from claims databases and/or EHRs) and a drug D

Do: Determine if evidence exists that associates D with a previously unanticipated adverse event

Note that no specific associated ADE has been hypothesized, and there is a need to identify the event to be predicted. To my knowledge, ML has not been applied to this task before now. I seek to build a model that can predict which patients are on drug D using the data after they start the drug (left-censored) and also censoring the indications of the drug. If a model can predict which patients are taking the drug, there must be some combination of clinical experiences more common among patients on the drug. In theory, this commonality should not consist of common causes for use of the drug, but common effects. The model can then be examined by experts to see if it might indicate a possible adverse event.

4.2.1 Implementing Relational Learning for ADEs

To apply the relational learning-based ADE approach, I need to analyze in more detail:

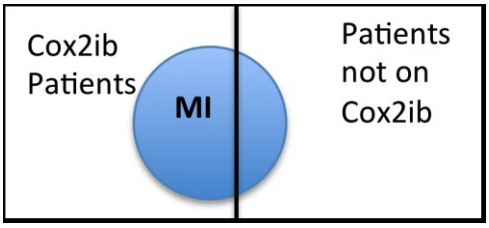
1. EHR data are multi-relational and temporal, necessitating relational learning (De Raedt 2008) for this task.
2. The output of the learning process should be easy to interpret by the domain expert (Page and Srinivasan 2003).
3. Generally, only a few patients on a drug D will experience novel ADEs (ADEs not already found during clinical trials). The learned model need not, and indeed most often should not, correctly identify everyone on the drug, but rather merely a subgroup of those on the drug while not generating many false positives (individuals not on the drug). This

argues that the learning problem actually can be viewed as "subgroup discovery"(Wrobel 1997; Klosgen 2002; Zelezn´y and Lavrac 2006), in this case finding a subgroup of patients on drug D who share some subsequent clinical events.

This suggests using a relational rule-based classifier, since relational rules naturally induce subgroups on the data, are discriminant, and are often easy to understand. In these experiments, the ILP system, Aleph (Srinivasan 2004) is used. In the remainder of the section, for concreteness, I present the discussion in terms of Aleph. Aleph learns rules in the form of Prolog clauses and scores rules by coverage *(POS-NEG)*, but this scoring function can be easily replaced by any user defined scoring function.

Suppose we did not know that Cox2ibs doubled the risk of MI, but we wondered if these drugs had any associated ADE. The relational ML approach can be seen as a retrospective cohort study, where positive examples are the patients on Cox2ibs and the negative examples are those individuals who have not used Cox2ibs. Choosing positive examples for training in supervised learning is fundamental in obtaining good study quality. There must be some evidence that the patient has the exposure of interest (Cox2ib). In addition, age and gender matched patients without Cox2ib are used as negative examples. In this case, for each positive example, a patient of the same age and gender who is not on a Cox2ib. (Negative examples could be selected to be similar to the cases in other ways—age and gender are just the most common such features in clinical studies.) Because Cox2ibs double the risk of MI, one can expect the

Figure 4.1: Distribution of people with risk of myocardial infarction (MI)



distribution of selected patients to appear as in figure 4.1. For example, if we have say 200 positive (POS) patients who suffer an MI, we expect about 100 negative (NEG) patients. The following rule would have a strong score of *POS* - *NEG* = 100 and hence would be returned by Aleph unless some other rule scores even better. The following rule says that a patient was likely on a Cox2ib if they suffered an MI.

cox2ib(Patient) ← *mi(Patient)*

Another advantage of the multi-relational approach is that the body (precondition) of the rule does not have to be a single condition, but it can be a combination of conditions and lab results, possibly in a temporal order. Hence, ADEs that do not neatly correspond to an exact pre-existing diagnosis code can be discovered. Furthermore, the body of the rule can involve other drugs. So, ADEs caused by drug interactions can be captured. For example, it has recently been observed that patients on Plavix may have an increased risk of stroke (ordinarily prevented by Plavix) if they are also on Omeprazole. This can be represented by the following rule:

plavix(Patient) ← *omeprazole(Patient)* ^ *stroke(Patient)*

Just because the rule is representable does not mean it will be learned. This depends on its support in the data, and the support of other rules that could score better, specifically as the support impacts the scoring function that is used.

4.3 EXPERIMENT WITH A REAL EHR AND KNOWN ADES

This section reports experiments using Aleph to identify previously unanticipated adverse events based on the analysis of clinical data obtained from the electronic medical records. Specifically, the study was performed to see if there were any unanticipated adverse events that occurred

when subjects used Cox2 inhibitors (Vioxx, Celebrex and Bextra).

4.3.1 Population

This study utilized the Marshfield Clinic's Personalized Medicine Research Project (PMRP) cohort (McCarty et al, 2005) consisting of approximately 19,700+ subjects. The PMRP cohort included adults aged 18 years and older who reside in the Marshfield Epidemiology Study Area (MESA). Except for the city of Marshfield, MESA residents reside rurally or in small towns or villages. There is very low annual in- and out-migration, which makes it ideal for prospective studies and the population is comparable to the Wisconsin population. Most of the subjects in this cohort receive most, if not all, of their medical care through the Marshfield Clinic integrated health care system. Marshfield Clinic has used an EHR since the late-1980s and thus, there is electronic clinical history on patients dating back several decades.

4.3.2 Data Source

Marshfield has one of the oldest internally developed EHRs (Cattails MD) in the US, with coded diagnoses dating back to the early 1960s. CattailsMD has over 13,000 users throughout Central and Northern Wisconsin. Data collected for clinical care is transferred daily into the Marshfield Clinic Data Warehouse (CDW) where it is integrated. The CDW is the source of data for this study. Programs were developed to select, de-identify by removing direct identifiers, and then transfer the data to a collaboration server. For this investigation, the specific CDW tables used were: ICD9 diagnoses, observations (including lab results and other items such as weight, blood pressure, and height), three sources of medication information and patient demographics

(gender and date of birth).

4.3.3 Data Pre-Processing

Prior to running Aleph the de-identified EHR database was converted into Prolog facts. Specifically, each row of each of the major CDW tables was converted into a Prolog fact consisting of Patient ID, Age at event, and the fact code and/or result (for labs and vitals). Examples of the clinical Prolog facts can be seen below. Age at event was calculated by subtracting the date of birth from the actual event date.

diagnoses('Patient1', 43.78, '414.0','Coronary atherosclerosis'). hasdrug('Patient1', 56.34, 'CEPHALEXIN MONOHYDRATE').
vitals('Patient1', 55.6, 'Systolic', 128).
gender('Patient1','Female').
dob('Patient1', 19610122).
observations('Patient1',31.159,'Urinalysis-Glucose','NEG').
observations('Patient1',31.159,'Urinalysis-Hyaline','0-2').
observations('Patient1',44.937,'White Blood Cell Count (WBC)',8.0).

For this analysis, medicated subjects were identified (referred to as positive examples) by searching through the medications database and selecting those subjects having an indicated Cox2ib use. An equal number of subjects (referred to as negative examples) were randomly selected from a pool that were matched to the cases based on gender and age. The negative examples could not have any indicated use of Cox2ib drug in the medications database.

4.3.4 Censoring

For machine learning negative examples were age and gender matched to positive examples. To learn rules that predict drug use based on events that occur after initiation of therapy, rather than reasons for the drug use (indications), the data was left-censored. Left

censoring positive examples involved removing all data about a patient before the date he or she began using the drug. Negative examples were censored in the same way, or else ML might learn a rule to distinguish positive examples from negative examples simply based on the presence of data from twenty years ago. But negative examples by definition do not have a date at which they began the drug. If we use a specific date for all of them, such as the median date for drug initiation by positive examples, again ML can find useless rules based on this artifact. My approach is to match each negative example with a positive example, based on age and gender, and censor both at the same date.

4.3.5 Scoring Function and Filtering

By running the Relational ML approach with censoring by matching on EHR data, I got a plethora of rules that simply capture indications for the drug. This is because for many patients, the reason they take the drug still continues to get entered into the EHR after drug initiation. For example, patients who go on Cox2 inhibitors (Cox2ib) for arthritis continue to have diagnoses of "arthritis" entered into the EHR after they initiate Cox2ib therapy. This may occur because the problem continues or because the physician continues to refer to the earlier diagnosis in recording new visits with the patient. To address (though not completely solve) this problem, the scoring function was modified to be time-based. By default, the ILP system Aleph (Srinivasan, 2004) uses the scoring function coverage as (POS – NEG), where POS is the number of positive examples covered by a rule and NEG is the number of negative examples covered by the rule. In this application, and other applications that use longitudinal or temporal data such as clinical histories, rules that have a higher P − N score in data after a particular event (drug initiation in

this case) are preferred over the P-N score before the event, is preferred. Hence the scoring function was modified to be $(POS_{after} - NEG_{after}) - (POS_{before} - NEG_{before})$. Thus, for each patient, there is a version of the background knowledge about that patient that is left-censored (for the "after" scoring) and a version that is right-censored (for the *before* scoring).

All records for each positive example (patient on Cox2ib) were identified as either *before* or *after* records, by whether or not their dates preceded the initial Cox2ib prescription; records for each negative example were similarly divided, but based on the Cox2ib initiation date for the corresponding case (since a control has no Cox2ib initiation date). The revised scoring function that was employed within Aleph is $(POS_{after} - NEG_{after}) - (POS_{before} - NEG_{before})$. In addition, the records were limited to within 5 years of the censor date both in the before and after versions of the data.

4.3.6 Validation

Because of the size and complexity of EHR data, as well as the significant runtime for ILP on such data, instead of cross-validation, the validation was done by a single train-test split. Seventy five percent of the data was used for training, while the remaining 25% was used for testing.

4.3.7 Results

There were 19,660 subjects within the PMRP cohort that had medication records. Table 4.1 shows the 10 most significant rules identified by Aleph for a single run. Note that the penultimate rule (highlighted) identifies the diagnosis of 410 (MI) as a possible ADE of Cox2.

The fact that this ADE can be learned from data suggests that this method is capable of identifying important drug interactions and side-effects.

Table 4.1: Top 10 Most Significant Diagnoses Identified for Cox2 Medication Use

Rules for Cox2(A) :-	Pos	Neg	Total	P-value
diagnoses(A,_,'790.29','Abnormal Glucose Test, Other Abn Glucose',_).	333	137	470	6.80E-20
diagnoses(A,_,'V54.89','Other Orthopedic Aftercare ',_).	403	189	592	8.59E-19
diagnoses(A,_,'V58.76','Aftcare Foll Surg Of The Genitourinary Sys',_).	287	129	416	6.58E-15
diagnoses(A,_,'V06.1','Diphtheria-Tetanus-Pertussis,Comb(Dtp)(Dtap)',_).	211	82	293	2.88E-14
diagnoses(A,_,'959.19','Other Injury Of Other Sites Of Trunk ',_).	212	89	301	9.86E-13
diagnoses(A,_,'959.11','Other Injury Of Chest Wall',_).	195	81	276	5.17E-12
diagnoses(A,_,'V58.75','Aftcar Foll Surg Of Teeth, Oral Cav, Dig Sys',_).	236	115	351	9.88E-11
diagnoses(A,_,'V58.72','Aftercare Following Surgery Nervous Syst, Nec',_)	222	106	328	1.40E-10
diagnoses(A,_,'410','Myocardial Infarction',_).	212	100	312	2.13E-10
diagnoses(A,_,'790.21','Impaired Fasting Glucose ',_).	182	80	262	2.62E-10

In some cases a drug may cause an ADE that does not neatly correspond to an existing diagnosis code (e.g., ICD9 code), or that only occurs in the presence of an- other drug or other preconditions. In such a case, simple 1-literal rules will not suffice to capture the ADE. I now report a run in which all of the background knowledge was used, including labs, vitals, demographics and other drugs. Table 4.2 shows the top ten most significant rules. The use of ILP yields interpretable rules. Fisher's exact test indicated that many rules demonstrated a significant difference in identifying positive cases over chance. The sobering aspect of this result is that Aleph learns over a hundred rules, but most appear to simply describe combinations of features associated with indications for the drug. A clinician would be required to sort through this large set of rules in order to find any evidence for possible ADEs. Further research is required to find ways to reduce the burden on the clinician, including automatically focusing the rule set toward possible ADEs and presenting the remaining rules in a manner most likely to ease human effort.

Table 4.2: Top 10 Most Significant Rules Identified for Cox2 Medication Use - Rules used EHR background data including diagnoses, medications, laboratory observations, demographic and procedure files. The rules are presented in a human interpretable format.

Rules for Cox2(A) :-	Pos	Neg	Total	P-value
gender(A,'Female'), hasdrug(A,_,'IBUPROFEN'), diagnoses(A,_,'305.1','Tobacco Use Disorder',_).	509	177	686	4.24511E-38
diagnoses(A,B,'462','Acute Pharyngitis',_), hasdrug(A,B,'IBUPROFEN').	457	148	605	1.27208E-37
hasdrug(A,_,'NORGESTIMATE-ETHINYL ESTRADIOL'), gender(A,'Female').	339	88	427	8.11776E-36
diagnoses(A,_,'V70.0','Routine Medical Exam',_), hasdrug(A,B,'IBUPROFEN'), diagnoses(A,B,'724.2','Lumbago',_).	531	199	730	1.00157E-35
diagnoses(A,_,'462','Acute Pharyngitis',_), gender(A,'Male').	433	144	577	1.44421E-34
diagnoses(A,_,'89.39','Nonoperative Exams Nec',_), diagnoses(A,_,'305.1','Tobacco Use Disorder',_).	502	186	688	2.01546E-34
hasdrug(A,_,'CYCLOBENZAPRINE HCL'), gender(A,'Male').	415	135	550	4.12122E-34
hasdrug(A,_,'FLUOXETINE HCL'), gender(A,'Female').	493	189	682	3.59643E-32
l_observations(A,B,'Calcium',9.8), diagnoses(A,B,'724.5','Backache Nos',_).	487	189	676	3.28104E-31
diagnoses(A,_,'V71.89','Observ For Other Specified Suspected Condi 10/00',_), gender(A,'Male').	492	193	685	5.35027E-31

4.4 CONCLUSION

This chapter presents an initial study of machine learning for the discovery of unanticipated adverse drug events (ADEs). Machine learning shows promising results when it is used in a manner that might at first seem "in reverse," when the value of interest—in this case, some unanticipated ADE—is not known. My work shows that this approach is able to successfully uncover an ADE for Cox2 inhibitor.

This chapter demonstrates the importance of learning from other disciplines when selecting positive and negative examples for machine learning, as well as in setting the scoring function. I don't want to find patterns in the patients who get prescribed a particular drug, because that is already known—they are the indications of the drug. Hence, it is important to

control for drug indications by using data about patients before the drug so as to remove the indications from the data following drug use.

Another lesson is that despite the censoring, a high accuracy, or highly accurate discovered subgroup, does not automatically mean we have uncovered one or more ADEs. Instead, all rules must be vetted by a human expert to determine if they are representative of an ADE or of some other phenomenon, such as a patient on arthritis medication such as Cox2ib also suffers from other correlated ailments. Once these associated conditions are also censored, learning ideally should be re-run in case ADEs were masked by other rules that scored better.

Another lesson is that data are multi-relational, including longitudinal (temporal), and hence may be best analyzed by methods that can directly handle such data. It would be desirable to take into account time from drug exposure to events, but this is a challenging direction because different drugs can cause ADEs over different ranges of time. Some drugs may cause an ADE within hours after they are taken, whereas others may have permanent effects that only manifest themselves as an ADE years later.

4.5 APPLICATIONS FOR MACHINE LEARNING IN ACTIVE SURVEILLANCE

In addition to the task of ADE that I have presented, machine learning approaches could support many drug safety needs, including:

1. *Identify and characterize temporal relationships between drugs and conditions across the population* - Is there an association between exposure to rofecoxib and cardiovascular events such as MI? If so, what is the likely time-to-onset of the event, relative to exposure? Does the risk increase over time and vary by dose?

2. *Identify drug-condition relationships within patient subpopulations* - Among elderly, what are the observed effects of a particular medicine? Among patients with renal impairment, what is rate of adverse events?
3. *Identify drug-drug interactions that produce harmful effects* - Which concomitant drug combinations produce elevated risks, relative to exposure to individual products?
4. *Identify risk factors and define patient subgroups with differential effects of a drug-related adverse event* - Which patients are more likely to experience adverse events? Which patients less likely to experience adverse events?
5. *Create models for predicting event onset* - Which patients are likely to have experienced a MI, based on available information about diagnoses (AMI and other CV terms), diagnostic procedures (EKG), treatments (PCI), lab tests (troponin, CK-MB), and other observations.

Identifying previously unanticipated ADEs, predicting who is most at risk for an ADE, and predicting safe and efficacious doses of drugs for particular patients all are important needs for society. With the recent advent of "paperless" medical record systems, the pieces are in place for machine learning to help meet these important needs.

4.6 CONTRIBUTORSHIP TO THIS CHAPTER

This chapter represents work done by Peggy Peissig. Other contributors to this research include David Page and Vitor Santos Costa who provided oversight for study design, relational learning and scoring function development. Michael Caldwell provided clinical interpretation of ILP rules. Aubrey Barnard, Peggy Peissig and Sriraam Natarajan contributed experiments (using

relational machine learning with different scoring functions, censoring methods and data), to a paper entitled *Identifying Adverse drug Events by Relational Learning* (Page et al, 2012). This paper was a collection of approaches using Relational ML for ADE detection. All authors read and approved the final published manuscript.

CHAPTER 5

Relational Machine Learning for Electronic Health Record-Driven Phenotyping

As already noted in the Chapter 1, developing phenotyping algorithms that use EHR data currently requires medical insight, which is based on the perceptions of clinical experts. In the current age of big data, experts rarely have the time to look at all the data for more than a few patients, which only represent a small fraction of those in the EHR. This chapter introduces methods to minimize reference expert effort, time and even to some extent possible expert biases in EHR-based phenotyping. While machine learning has been used for phenotyping over the last two years, as I discuss in the next section, the novel components of this chapter are to compare the relational approach against simpler feature vector-based approaches used previously and most importantly to avoid the need for human-labeled training data. Specifically, building on the previous chapter, this chapter presents an approach to automatically label training examples using ICD-9 diagnosis codes for relational learning and tests how well this works on 10 different phenotypes. In addition, the practice of infusing negative examples with borderline positive examples is used to improve model accuracy and interpretation. The relational learning phenotyping models are compared to two popular decision tree and rule-based machine learning approaches.

The novel use of relational learning to identify patients for research using EHR data supports my thesis that phenotyping can be done accurately and efficiently using computational methods.

5.1 BACKGROUND AND CONTRIBUTIONS

As noted in earlier chapters, in medical and pharmacogenetic research that attempts to identify and quantify relationships between exposures and outcomes, a critical step is the classification of subjects or phenotyping (Wojczynski and Tiwari, 2008; Rice, Saccone and Rasmussen, 2001; Gurwitz and Pirmohamed, 2010; Samuels, Burn and Chinnery, 2009). With the adoption of electronic health records (EHRs), millions of data points are becoming available (Linder et al, 2007) for use in EHR-driven phenotyping, a process whereby patients are electronically classified using EHR data. The eMERGE Network (Kho et al, 2011; McCarty et al, 2011) has successfully demonstrated the feasibility of EHR-driven high throughput phenotyping for identifying large numbers of research subjects for genome-wide association studies (Kho et al, 2012; Denny et al, 2011; Peissig et al, 2012; Jeff et al, 2013; Rasmussen-Torvik et al, 2012; Liao et al, 2013; Ritchie et al, 2013; Crosslin et al, 2012). The process used is largely dependent on multiple iterations of selecting patients from the EHR and then manually reviewing them to identify classification features (Newton et al, 20113). This approach is time-consuming (Wojczynski and Tiwari, 2008) and relies on expert perceptions and intuition. Experts have limited time and can only carefully examine a small fraction of the available EHR data for phenotype model development. With the enormous volume of data found in the EHR, experts may lose the ability to uncover “hidden” relationships or “unseen” attributes that are relevant to the phenotype. The result is a serious temporal and information bottleneck in the construction of high quality phenotypes for this type of research.

Machine learning (ML) has been introduced as an alternative phenotyping strategy, using structured data from the EHR for rheumatoid arthritis Carrol, Eyler and Denny, 2011), asthma (Anand and Downs, 2011), colorectal cancer (Xu et al, 2011), poorly controlled diabetes (Huang

et al, 2007), type 2 diabetes (Dingcheng et al, 2013) and heart failure (Pakhomov and Weston, 2007; Wu, Roy and Stewart, 2010) studies. These ML studies applied a variety of classical or rule-based ML approaches that take advantage of data placed into a fixed length feature table for analysis (Anand and Downs, 2010; Xu et al, 2011; Huang et al, 2007; Dingcheng et al, 2013; Pakhomov and Weston, 2007; Wu, Roy and Stewart, 2010). Development of the feature table relies on input from experts (physician) and/or the availability of existing manually validated "gold-standard" classified subjects to guide the supervised learning activity. The expert input is often flavored by existential perceptions (Van Sickle et al, 2013).

The most closely related work comes from Dingcheng *et al*, in phenotyping type 2 diabetes (Dingcheng et al, 2013). This work is similar to mine in that it also uses a rule-based ML approach, specifically the Apriori association rule learning algorithm. Hence, it shares with our approach the advantage of learning rules for phenotyping that are easily understood by human users. The primary difference between the approaches is the use of *relational learning* to directly learn from the tables of the EHR versus Apriori, which must learn from data conflated into a single table.

As noted in Chapter 2, ILP algorithms typically are employed to build models that predict future outcomes. This research applies several novel techniques to adapt ILP for the phenotyping task. The techniques employed in this chapter are novel to the young field of ML-based phenotyping and include generating training sets without expert (physician) involvement to provide supervision for the learning activities, left-censoring of background data to identify subgroups of patients that have similar features denoting the phenotype and infusing negative examples with borderline positive examples to improve rule prediction.

5.2 MATERIALS AND METHODS

The Marshfield Clinic Research Foundation's Institutional Review Board approved this study. The goal of this research was two-fold: (1) to test the utility of *relational learning* for EHR-driven phenotyping; and (2) to develop methods that reduce expert time and enhance attribute awareness in the EHR-driven phenotyping process. Methods presented in this section were applied to ten phenotypes to demonstrate the generalizability of this approach for phenotyping. Using cataract as an example, I have presented a detailed description of these methods in Appendix B and made available examples of record formats and scripts in Appendix C and ILP rules in Appendix D. Figure 5.1 provides an overview of study data management for the various analyses.

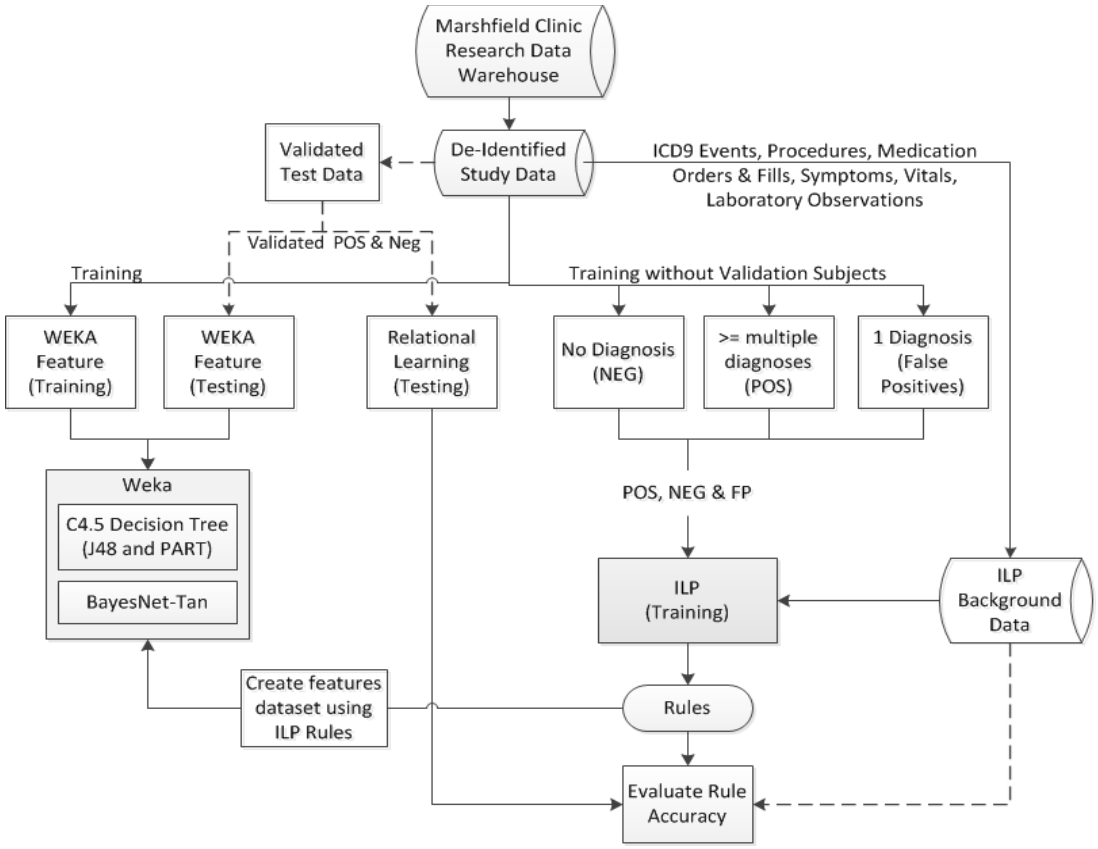


Figure 5.1: Overview of data preparation and analysis processes. Patient data from the data warehouse is de-identified and validation subjects are removed. Left side of figure shows data preparation for J48 and PART (WEKA) analyses. Right side of figure shows steps to identify training examples and integration of background data for the induction logic programming (ILP). Validation subjects are used for testing accuracy of Rules. Rules are

used to create features for WEKA Bayes-net Tan for creation of area under the receiver operator characteristic curve.

5.2.1 Data sources and study cohort

The research data warehouse for Marshfield Clinic's internally developed CattailsMD EHR was the source of data for this investigation. Medical events, also referred to as data sources, included diagnoses, procedures, laboratory result values, medications, biometric measures, and symptoms recorded in CattailsMD for each patient visit. This EHR data on patients residing in a 22 ZIP Code area was selected from the research data warehouse and de-identified for use in this study.

5.2.2 Phenotype selection

Phenotypes were selected based on one of two criteria: (1) a pre-existing manually validated cohort was available to test the accuracy of the EHR-driven phenotyping model (Peissig et al, 2012; Berg et al, 2010; Kawaler et al, 2012); or (2) the phenotype was an adverse drug event (ADE) outcome required for other research. The evaluation was performed on expert labeled data (referred to as validates or test datasets). The training was performed on subjects identified using an automated procedure that will be described in the next section.

5.2.3 Identifying training examples for supervised learning

Identifying training examples for supervised learning is a critical task. Several ML studies have used experts (physicians) to review medical records to classify patients as either cases (positive examples) or controls (negative examples) (Carroll et al, 2011) or used pre-existing validated cohorts to construct training sets to guide the supervised learning activity. A goal of this research was to develop methods to reduce expert time required for EHR-driven

phenotyping, thus if I could develop an approach for selecting training examples without physician input, that would be optimal.

A recent study by Carroll *et al.* indicated that ICD-9 CM diagnostic codes were an important feature when using support vector machine learning to characterize research subjects (Carroll et al, 2011). This knowledge coupled with past experience, prompted the use of ICD-9 diagnostic codes to identify potential positive (POS) examples for the supervised learning task. So, for training the models, patients having an excessive numbers of diagnoses were labeled as POS. Patients without a diagnosis were labeled as negative examples (NEG), and patients with only a single diagnosis or multiple diagnoses on the same day were labeled as false positives (FPs). Refer to table 5.1 for the exact number of diagnoses used to select patients to train for each phenotype.

5.2.4 Relational Learning and ILP approach

Traditional *relational learning* use in the medical domain has focused on predicting patient outcomes. Supervision for the prediction task came from patient examples representing positive or negative outcomes, given some common exposure. For example, to develop a model that will predict diabetic retinopathy, given a patient has diabetes; the supervision comes in the form of POS (diabetic patients that have diabetic retinopathy) and NEG (diabetic patients without diabetic retinopathy). The EHR data collected before the diabetic retinopathy occurrence is used to build a model to predict whether a diabetic patient is at future risk for diabetic retinopathy (refer to figure 5.2.A). *Relational learning* applied to the phenotyping task uses a similar approach but in a reversed fashion, meaning that we use EHR data after the diabetic retinopathy occurrence to identify common attributes that patients with diabetic retinopathy have in common before they develop the disease (refer to figure 5.2.B). The supervision comes from

patients with diabetic retinopathy (POS) and patients that do not have diabetic retinopathy (NEG).

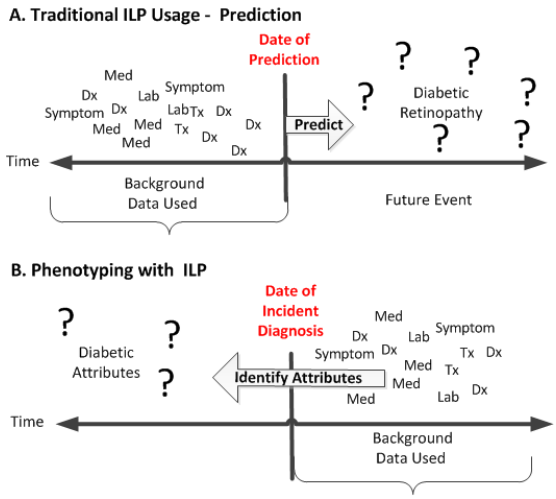


Figure 5.2: (A) Inductive logic programming (ILP) uses data collected prior to a prediction date to predict disease outcomes. (B) Phenotyping using ILP uses data collected after the incident date (of a condition), to predict features that a subgroup may be sharing that are representative of a phenotype.

When phenotyping, we should not assume that we know in advance all the clinical attributes a particular phenotype might have that could be used to succinctly identify it. Suppose we do not know in advance that diabetes is associated with elevated blood sugar. The POS and NEG cannot be divided based on elevated blood sugar, because it is not yet known that elevated blood sugar is an indicator for diabetes. Instead, the problem can be addressed by running ILP using data after the first diagnosis to detect attributes or conditions that can be used to identify who has diabetes. This seems counter-intuitive, because we are training on data obtained after we already know who is a diabetic, but if we can predict diabetic patients based on similar medical attributes existing prior to diagnosis, we may be able to uncover unknown (unbiased) attributes that define the phenotype.

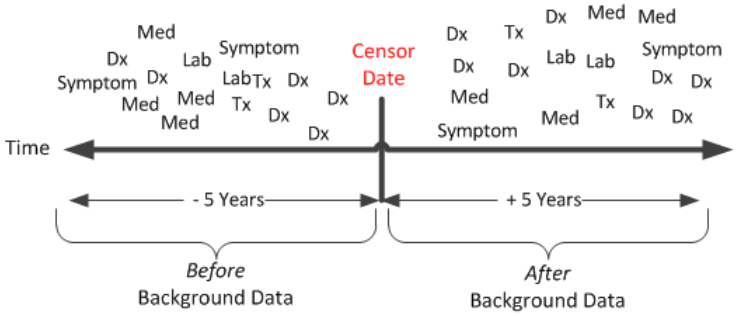
5.2.4.1 Constructing background knowledge

Detailed methods surrounding background file creation can be found in Appendix B-3 are summarized here. Coded ICD-9 diagnosis, medication, laboratory, vitals, symptoms, and procedure records from the EHR were used as "background knowledge" for model building. A censor date was determined for each POS and FP based on the initial diagnosis date of the phenotype. Records were labeled as *before* or *after* based on their relationship to the censor date. *Before* records occurred before the censor date and *after* records occurred after the censor date. Records for each NEG were similarly divided but based on the censor date of the corresponding POS (since NEGS have no incident diagnosis date).

5.2.4.2 ILP Scoring functions:

Scoring functions in Aleph and other ILP systems evaluate the quality of a rule, and thus, are fundamental to the learning process. I tested three different scoring functions with Aleph (Muggleton, King and Sternberg, 1992). The first function was ($POS_{(after)} - NEG_{(after)}$), where $POS_{(after)}$ denotes positive examples that use EHR data *after* the censor date and $NEG_{(after)}$ denotes negative examples that use EHR data *after* the censor date (figure 5.3). The second function was ($POS_{(after)} - (NEG_{(after)} + POS_{(before)})$), in which $POS_{(before)}$ denotes positive examples that use EHR records *before* the censor date and $POS_{(after)}$ and $NEG_{(after)}$ are analogous with the previous example. Early on, I found that the later scoring function improved model accuracy and thus, used it for this study. Henceforth, I will refer to this function as ILP.

Figure 5.3: Censoring data to support inductive logic programming scoring functions



From previous work, I found that using the ILP scoring function tended to create rules that could differentiate the POS and NEGs based on ICD-9 codes relatively well but often failed to provide more specific rules that could discriminate borderline POS and NEG examples. To further discriminate and improve accuracy of the rules, we infused the NEG examples with subjects that we considered to be false positivies. A false positive is defined as patients having a single diagnosis or multiple diagnoses on the same day. In other words, we are not sure if the patient does or does not have the condition because some diagnostic codes are entered to substantiate ordering a test and may not be valid diagnoses. These are examples that have one relevant diagnosis code, but not two; or several diagnoses on the same day with no subsequent follow-up; while not all of these are guaranteed to be FPs; they are enriched for false positives

and, therefore, should increase precision of learned rules. The scoring function to support the addition of false positives is: ($POS_{(after)} - (NEG_{(after)} + POS_{(before)} + False\ Positives_{(after)})$). Henceforth, this function will be referred to as ILP+ false positives.

I used the Aleph ILP system, based on Muggleton's Progol algorithm (Muggleton and King, 1992) running under YAP Prolog as the machine learning (ILP) approach in this research. Aspects of file preparation are found in Appendix B-4. Examples of configuration files can be found in Appendix C (cat.b – file which contains the parameters for running Aleph and runAleph.sh – a script that initiates the Aleph phenotype model building session).

5.2.5 Classical Machine Learning Approaches

The same training and validation sets used for *relational learning* were also used for the classical ML approaches. I limited features to unique facts found in the data sources. Features were selected if they were shared by more than 0.25% of the training subjects. Frequency counts for each unique feature, by data source, were combined into a single record for each patient. All training set patient records were combined into a single "training" file for analysis. A test file was developed using validation sample patients. The same features identified by the training set were used for the validation patients when constructing unique patient feature records.

In order to make a comprehensive comparison, I compared the aforementioned ILP models to two popular ML classifiers available in the widely used WEKA software (WEKA). J48 is based on a Java implementation of the well-known decision tree classifier C4.5 (Quinlan, 1993) and PART, a rule based classifier based on Classical and Regression Tree (Breiman et al, 1984). Both of these implementations were available in WEKA, a widely used suite of ML algorithms.

5.2.6 Validation

Phenotype models were evaluated against independent, manually validated cohorts obtained from previous research (Peissig et al, 2012; Berg et al, 2010; Kawaler et al, 2012) or developed for this purpose. The validated subjects were removed from the sampling frame prior to the selection of POS, NEG, and FPs used in the training set for model building activities. The validation cohort consisted of POS and NEG examples. Two cohorts (congestive heart failure [CHF] and acute liver injury [ALI]) were constructed by randomly selecting patients using ICD-9 diagnoses codes (cases) or the absence of the ICD-9 code (controls). A trained research coordinator verified the presence or absence of the phenotype for the ALI and CHF validation sets. A second research coordinator reviewed 10% of the records (five records each for CHF and ALI); a board-certified physician resolved disagreements or questions surrounding the classifications for three records in ALI. Validation cohorts obtained from previous studies were manually validated in those respective studies.

5.2.7 Analysis

I selected to report accuracy, precision, recall, F-measure (defined as 2 x [(recall x precision) / (recall + precision)] and whenever possible, area under the receiver operator characteristic (AUROC) curve, for each model (ILP, ILP+ false positives, PART and J48) by phenotype (Carroll, Eyler and Denny, 2011; Anand and Downs, 2010; Xu et al, 2011; Huang et al, 2007; Pakhomov and Weston, 2007). To associate probabilities with ILP and ILP+FP classifications for AUROC, I used the Bayes Net-TAN classifier as implemented in WEKA version 3.6.9 for each phenotype (Ho et al, 2012). Such use of a Bayesian network to combine relational rules learned by ILP is a popular approach used in statistical relational learning (Getoor and Taskar, 2007). Performance measurements were calculated using the number of

correctly classified validation set subjects. Significance testing using a two-sided sign test (binomial distribution test) at 95% confidence, was used to evaluate model sensitivity when adding varying levels of false positives to the ILP+ false positives scoring function and to evaluate a difference in overall model performance between any of the ILP, ILP+ false positives, PART and J48 models.

5.3 RESULTS

The sampling frame used for all phenotype-modeling activities consisted of 113,493 subjects. Table 5.2 provides descriptive information on the phenotype validation datasets. Overall testing was done using 1471 positive examples and 1383 negative examples. Ten models were learned with the validation results appearing in table 5.3. Asthma, type 2 diabetes, diabetic retinopathy, and deep vein thrombosis phenotype models had the highest ILP accuracy measurements (>0.94), with acute liver injury and atrial fibrillation models having the lowest. Area under the receiver operator characteristics (AUROC) curve can be found for 8 of the phenotypes in figure 5.4. Diabetes was not displayed because it was very similar to the diabetic retinopathy phenotype.

The addition of FP examples had the desired effect of increasing precision but at the cost of decreased recall. There was no significant difference in overall model performance when adjusting the number of FPs (between 25%, 50%, 75%, and 100% of the POS examples) when using the ILP+FP scoring function. I found that adding FP examples to the scoring function yielded more descriptive rules for all phenotypes. Figure 5.5 provides an example of the rules that were created from ILP+FP model-building activities for acute liver injury.

The combined phenotype validation results are presented in table 5.4. There was no significant difference in overall accuracy between ILP and ILP+FP models, although ILP

performed significantly better than ILP+FP in detecting POS examples (p=0.0006), and ILP+FP performed significantly better than ILP when detecting NEG examples (p=0.008). When comparing ILP+FP to PART and J48, ILP+FP performed significantly better than PART (p=0.039) and J48 (p=0.003) for AUROC. There was no difference between the methods when evaluating accuracy and F-Measure.

A secondary study evaluating how well ILP+FP performed when compared to a common phenotyping practice called Rule-of-N was also conducted. Rule-of-N (usually Rule-of-2) is defined as an individual having 2 or more diagnoses (of the phenotype being considered), from a physician on two separate days. Each phenotype validation cohort was used to identify patients who met the rule of two criteria (both positive and negative examples). I then compared the correctly classified patients from Rule-of-2 to the ILP+FP results. When comparing the ILP+FP to the Rule-of-2 approach, ILP+FP outperformed the Rule-of-2 model in overall accuracy (p= 1.513E-8) and when identifying positive (p= 6.72E-28) examples. Rule-of-2 outperformed ILP+FP when identifying negative examples (p= 5.00E-09).

Table 5.1: Phenotypes and sampling frame

Phenotypes	ICD-9 Diagnosis Codes used for Training Set Identification	Minimum # ICD-9 Codes used to select Positives	Pool of available Positives[1]	# Training Positives	# Training Negatives	# Training False Positives
Acute Myocardial Infarction	410.*	20+	4364	1500	1500	460
Acute Liver Injury	277.4, 572.1-4, 573.1-4, 576.8, 782.4, 782.8, 790.40	15+	7393	314	314	314
Atrial Fibrillation	427.31	20+	6619	1000	1000	489
Asthma	493.*	20+	17579	1000	1000	1000
Cataracts	366.00-366.9 & 743.30 - 743.34	20+	19150	1000	1000	1000
Congestive Heart Failure	428*	20+	8280	1000	1000	750
Dementia	290.00, 290.10, 290.11, 290.12, 290.13, 290.3, 290.20, 290.0, 290.21, 291.2, 292.82, 294.1, 294.11, 294.10, 294.20, 294.21, 331.19, 331.82, 290.40, 290.41, 290.42, 290.43	20+	4139	1126	1126	657
Type 2 Diabetes	250.*	20+	10899	1500	1500	1000
Diabetic Retinopathy	362.01, 362.10, 362.12, 362.82, 362.84, 362.85, 362.07, 362.02, 362.03, 362.04, 362.05, 362.06	20+	2525	606	606	606
Deep Vein Thrombosis	453.*	20+	4140	1000	1000	658

* = include all decimal digits

[1]Includes all patients with at least one ICD-9 diagnosis code.

Note: Phenotype models were constructed for ten conditions. Training positive and false positive examples were identified using ICD-9 diagnosis codes. Negative training examples had no ICD-9 diagnosis code.

Table 5.2: Validation sample characteristics

Phenotypes	Total Number		Female (%)	Mean Age[3]	Years Followup[4] (StDev)		ICD-9 Diagnosis Count (StDev)	
	POS[1]	NEG[2]			POS[1]	NEG[2]	POS[1]	NEG[2]
Acute Myocardial Infarction	363	158	199 (38.2%)	73.8	37.9 (9.7)	33.5 (12.2)	1848 (1475)	1384 (1304)
Acute Liver Injury	44	6	23(46%)	69.3	34.9 (10.6)	35.2 (10.0)	1880 (1568)	2550 (951)
Atrial Fibrillation	36	35	31 (44%)	79.8	29.2 (13.1)	31.7 (14.5)	1345 (811)	1468 (1100)
Asthma	51	-	31 (61%)	61	34.2 (13.7)	-	1322 (814)	-
Cataracts	244	110	210 (59.32%)	75.2	39.7 (9.8)	37.9 (11.2)	1395 (1004)	864 (616)
CHF	60	36	51 (52%)	70	35.4 (10.0)	26.1 (13.8)	1614 (1184)	623 (647)
Dementia	303	70	203 (54.4%)	84	36.9 (11.3)	37.4 (8.64)	1579 (980)	1438 (1067)
Type 2 Diabetes	113	52	99 (60%)	67	36.3 (12.3)	34.0 (13.6)	1447 (781)	925 (781)
Diabetic Retinopathy	40	46	39 (45.4%)	71.6	35.1 (12.7)	37.9 13.8)	2032 (1158)	1614 (1158)
Deep Vein Thrombosis	217	870	614 (56%)	76	38.3 (9.9)	38.9 (10.2)	1947 (1604)	1269 (836)

[1]POS indicates Positive examples
[2]NEG indicates Negative examples
[3]Mean age calculated by (Year of data pull (2012) - Birth Year)
[4]Years Follow-up calculated by determining difference between first and last diagnosis date
Note: Phenotype models were validated using these validation cohorts.

Table 5.3: Phenotype model validation results by phenotype. This table presents a comparison of Machine Learning methods for comparison.

	PHENOTYPE									
	Acute Myocardial Infarction	**Acute Liver Injury**	**Atrial Fibrillation**	**Asthma**	**Cataracts**	**CHF**	**Dementia**	**Type 2 Diabetes**	**Diabetic Retinopathy**	**Deep Vein Thrombosis**
Accuracy										
ILP[1]	0.800	0.600	**0.775**	1.000	0.890	0.865	0.810	0.939	0.977	**0.980**
ILP+FP[2]	**0.810**	0.640	0.732	1.000	**0.898**	**0.885**	0.810	**0.945**	**0.988**	0.965
PART[3]	0.775	0.660	**0.775**	1.000	0.879	0.875	**0.850**	**0.945**	**0.988**	0.949
J48[4]	0.785	**0.700**	**0.775**	1.000	0.822	0.875	0.834	**0.945**	**0.988**	0.949
Precision										
ILP[1]	0.863	**0.929**	0.700	1.000	0.865	0.831	0.858	0.919	0.952	**0.990**
ILP+FP[2]	0.877	0.906	0.689	1.000	**0.877**	0.855	**0.936**	0.926	0.976	0.954
PART[3]	0.790	0.848	**0.824**	1.000	**0.877**	0.811	0.859	**0.949**	**0.989**	0.949
J48[4]	**0.797**	0.829	**0.824**	1.000	0.819	**0.881**	0.850	**0.949**	**0.989**	0.949
Recall										
ILP[1]	**0.850**	0.591	**0.972**	0.996	**0.996**	**0.980**	**0.858**	**1.000**	**1.000**	0.910
ILP+FP[2]	**0.850**	0.659	0.860	**1.000**	0.992	**0.980**	0.822	**1.000**	**1.000**	0.870
PART[3]	0.755	0.660	0.775	**1.000**	0.879	0.875	0.850	0.945	0.988	0.949
J48[4]	0.785	**0.700**	0.755	**1.000**	0.822	0.875	0.834	0.945	0.9888	**0.960**
F-Measure										
ILP[1]	0.856	0.722	**0.813**	**1.000**	0.926	0.900	0.858	0.958	0.976	0.947
ILP+FP[2]	**0.879**	**0.763**	0.795	**1.000**	**0.939**	**0.914**	**0.894**	**0.961**	**0.988**	0.940
PART[3]	0.788	0.719	0.765	**1.000**	0.877	0.871	0.854	0.944	**0.988**	0.949
J48[4]	0.789	0.747	0.765	**1.000**	0.820	0.871	0.840	0.944	**0.988**	**0.960**
AUROC[7]										
ILP+BNT[5]	0.769	**0.752**	0.772	n/a	**0.893**	0.825	0.817	0.904	**0.991**	0.953
ILPFP+BNT[6]	**0.831**	0.701	**0.774**	n/a	0.873	**0.914**	**0.831**	**0.957**	0.990	**0.971**
PART[3]	0.788	0.716	0.772	n/a	0.842	0.844	0.798	0.913	0.989	0.947
J48[4]	0.722	0.619	0.722	n/a	0.783	0.844	0.766	0.913	0.989	0.927

[1]ILPL: Inductive Logic Programming with using POS(after) - (NEG(after) + POS(before))
[2]ILP+FP: Inductive Logic Programming + False Positives using POS(after) - (NEG(after) + POS(before) + FP(after))
[3]PART: Java implementation of a rule based classifier in WEKA
[4]J48: Java implementation of C4.5 classifier available in WEKA
[5]ILP+BNT: BayesNet-Tan using ILP Classification Rules
[6]ILPFP+BNT: BayesNet-Tan using ILP+FP Classification Rules
AUROC: Area Under Receiver Operating Characteristics Curve **Bolded** numbers indicate highest score between phenotyping methods
Note: Phenotype model accuracy measurements were calculated for each scoring function by using the number of correctly classified positive and negative validation examples.

Table 5.4: Overall phenotyping approach evaluation.

	ILP[1]	ILP+FP[2]	PART[3]	J48[4]
# Positive Subjects	1471	1471	1471	1471
# Negative Subjects	1383	1383	1332	1332
Accuracy	0.900	0.894	0.886	0.883
F-Measure	0.900	0.896	0.888	0.885

[1]ILPL: Inductive Logic Programming with using POS(after) - (NEG(after) + POS(before))
[2]ILP+FP: Inductive Logic Programming + False Positives using POS(after) - (NEG(after) + POS(before) + FP(after))
[3]PART: Java implementation of a rule based classifier in WEKA
[4]J48: Java implementation of C4.5 classifier available in WEKA
Note: The results from a binomial classification (counting # wins for each method by phenotype), then using a two-sided sign test (binomial distribution test) at 95% confidence to determine if there is a difference. There was a significant difference favoring ILP+FP when compared to PART ($p=0.039$) and J48 ($p=0.003$) when evaluating AUROC. There was no significant difference when testing accuracy and F-Measure.

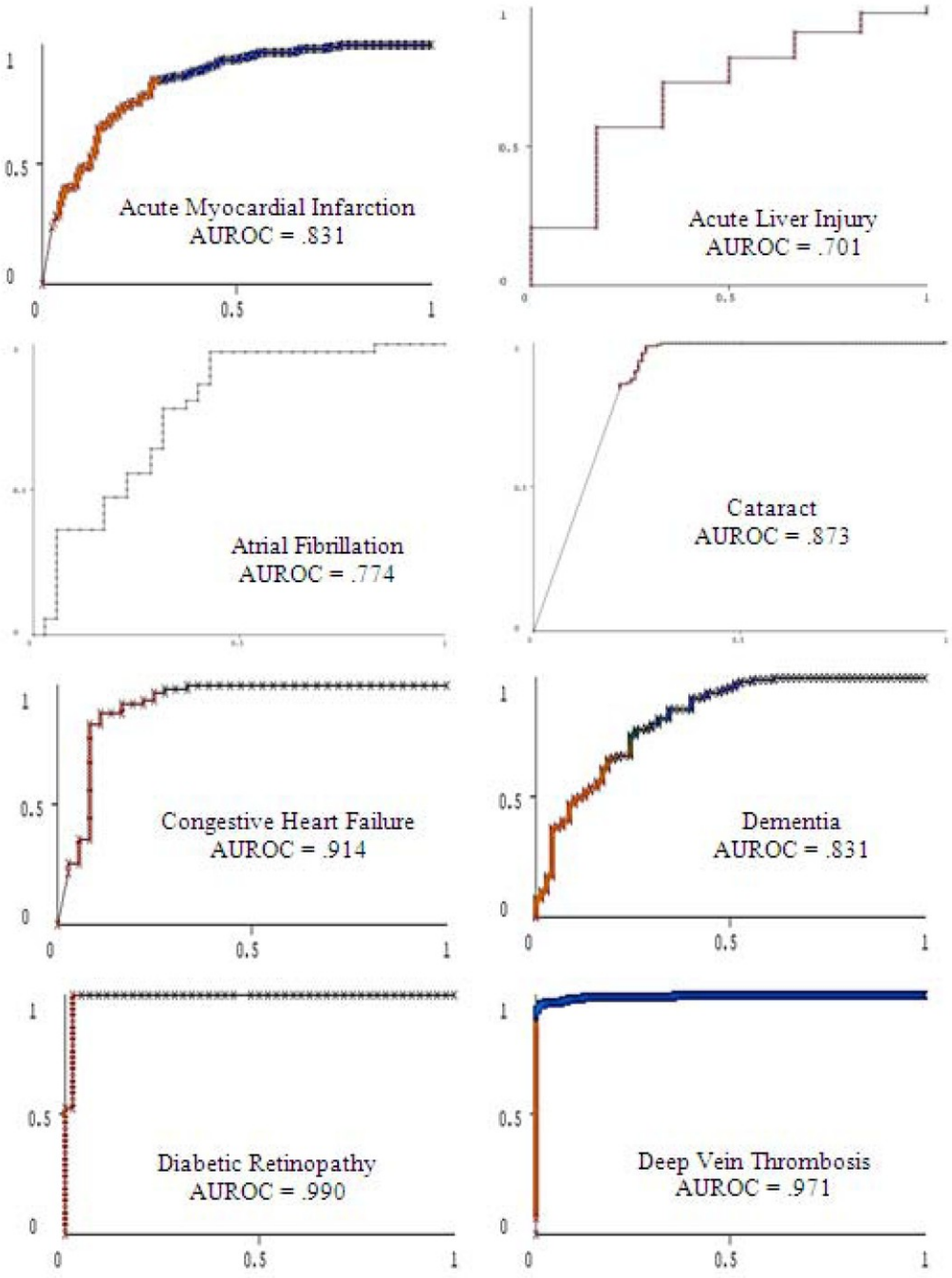


Figure 5.4: Area under receiver operator characteristic (AUROC) curves was used for selected phenotypes for ILP+FP. The diabetes AUROC curve is not displayed because it mirrors the diabetic retinopathy AUROC. Asthma is not displayed because negative examples were not available for calculations used to produce the AUROC curves.

Rule #	POS Cover[1]	NEG Cover[2]	Inductive Logic Programming Rule	Probability[3]
30	95	0	diagnoses(A,B,C,**'790.4','Elev Transaminase/Ldh'**,D), lab(A,E,20719,'Urea Nitrogen Bld',F,'Normal'), **lab(A,E,20727,'Alkaline Phosphatase (T-Alkp)',G,'High').**	1.00
35	52	0	has_tx(A,B,'99232','Sbsq Hospital Care/Day 25 Minutes',C,D,E,F), **lab(A,B,20816,'Differential Nucleated RBC',G,'High').**	1.00
42	129	0	**diagnoses(A,B,C,'782.4','Jaundice Nos',D), lab(A,E,20727,'Alkaline Phosphatase (T-Alkp)',F,'High').**	1.00
72	113	0	has_tx(A,B,'99214','Office Outpatient Visit 25 Minutes',C,D,E,F), **lab(A,G,20809,'Differential Segment Neut-Segs',H,'Normal'), lab(A,G,20728,'Bilirubin',F,'High').**	1.00
3	146	1	lab(A,B,20728,**'Bilirubin Total',C,'High'**), lab(A,D,20900,**'Direct Bilirubin',E,'High')**, lab(A,F,20857,'Red Cell Distribute Width(RDW)',G,'High').	0.99
51	142	1	**lab(A,B,20900,'Direct Bilirubin',C,'High'),** lab(A,B,20719,'Urea Nitrogen Bld',D,'Normal'), lab(A,B,20731,'AST (GOT)',E,'High').	0.99
11	138	1	**lab(A,B,20728,'Bilirubin Total',C,'High'),** lab(A,B,20809,'Differential Segment Neut-Segs',D,'Normal'), lab(A,E,20282,'Glucose',F,'High').	0.99
60	137	1	lab(A,B,20715,'Potassium (K)',C,'Normal'), **lab(A,B,20727,'Alkaline Phosphatase (T-Alkp)',D,'High'), lab(A,E,20901,'Unconjugated Bilirubin',F,'High').**	0.99

[1]Represents the number of positive examples covered by the rule

[2]Represents the number of negative examples covered by the rule

[3]Probatility = POS examples/(POSexamples + NEG examples)

Figure 5.5: A sample of the top "scoring" inductive logic programming rules for acute liver injury. The "bold" lettered rules are indicative of "facts" related to or associated with acute liver injury. The highlighted ILP rule (rule #35) represents a "fact" *(Differential Nucleated RBC' is 'High')* that was unknown to a physician reviewer prior to this investigation.

5.4 DISCUSSION

In this study, I used a de-identified version of the EHR coded data to construct phenotyping models for ten different phenotypes. This approach used: (1) ICD-9-CM diagnostic codes to define POS and NEG examples for the supervised learning task and (2) ILP to take

advantage of the relational structure of the EHR. ILP models were developed that produced F-Measures for seven of ten phenotypes that exceeded 0.90, which is comparable to other phenotyping investigations (Kho et al, 2012; Denny et al, 2011; Dzeroski et al, 2001; Ho et al, 2012; Kudyakov et al, 2012). For example, the diabetes phenotype was also studied by Dingcheng *et al*. (Dingcheng et al, 2013) where an F-measure of 0.914 was achieved; I achieved an F-measure of 0.926, albeit on different validation data.

Several of the phenotypes selected for use in this research (type 2 diabetes, diabetic retinopathy, dementia, and cataracts) corresponded to phenotypes used by the eMERGE network for genome-wide association studies. The eMERGE phenotyping models used a variety of EHR data, were developed using a multi-disciplinary team approach, and each phenotyping model took many months to construct. This method used similar coded EHR data, required minimum effort from the multi-disciplinary team, and developed phenotype models in a few days, however this development relied on pre-established validation cohorts. The phenotyping models were comparable in precision and recall when compared to eMERGE network algorithms (Kho et al, 2012; Peissig et al, 2012).

An advantage of using *relational learning* is that the ILP rules reflect characteristics of patient subgroups for the phenotype. These rules can be easily interpreted by a physician (or others), to identify relevant model features that not only identify patients, but also discriminate between patients that should or should not be classified as cases. In addition ILP rules are learned from the EHR database. These rules are not based on human intuition or "filtered" because of preconceived opinions about a phenotype. To emphasize this point, a physician

reviewer evaluated the ILP rules for acute liver injury in Figure 5.5 and questioned why high levels of "Differential Nucleated RBC' surfaced in Rule #35. After investigating a potential mechanism for the sudden rise in nucleated red cells, it is thought to be an injury to sinusoidal endothelium in bed, subject to ischemia as part of the fetal response to hypoxia or partial asphyxia, namely in the liver and bone marrow (Thilaganathan et al, 1994). Rule #35 provides an example that one's experiences, knowledge and/or possible bias may hide relevant information. After an investigation, this relevant information could be used to improve the phenotype model.

Initially, I used a simple scoring function that evaluated the differences between the POS and NEG examples using data captured *after* the initial diagnosis for both groups. I then tried to improve model accuracy by adding the *before* data for POS patients and *after* data for the FP patients; the goal of these additions was to mute some of the features that were common between true positive and false positive examples, thus making the model more discriminatory. Given the high recall and precision of this method in either case, only a few EHR-driven models yielded substantially different classifications between the two approaches, making it difficult to demonstrate that there is a difference in model performance when adding the $FP_{(after)}$ and $POS_{(before)}$ data. I speculate that larger phenotype validation sets may allow one to see a difference if it exists.

Inductive logic programming provides a series of rules that identify patients with a given phenotype. Most of the rules include a diagnostic code (because POS selection of training subjects was based on diagnostic codes) along with one or more other features. I noticed that in

some situations, ILP would learn a rule that was too general and, thus, invited the identification of FPs. Future research is needed to examine grouping of rules and selection of subjects based on a combination of rule conditions, thereby combining the advantages of ILP and the general "rule-of-N" approach commonly used in phenotyping which states a unique event must be present on "N" days to determine a case/control.

One limitation of this study is the solitary use of coded data found within the EHR (Carroll et al, 2011; Penz, Wilcox and Hurdle, 2007) for phenotyping. Other studies have indicated clinical narratives and images provide more specific information to refine phenotyping models (Peissig et al, 2012; Xu et al, 2011; Penz, Wilcox and Hurdle, 2007). I envision the use of natural language processing and optical character recognition techniques will increase the availability of structured data and thus improve *relational learning* model results as noted by Saria *et al* (Saria et al, 2010). More research is needed on this topic.

Another limitation is that a single EHR and institution was used in this research, thus limiting the generalizability of the study results. More research is needed to apply these methods across several institutions, EHRs and non-disease phenotypes.

5.5 CONCLUSION

I believe this research has the potential to address several challenges of using "Big Data" when phenotyping. First, I introduced and showed promising results for *relational learning* as a possible alternative method for phenotyping when using large relational databases. Second, I introduced novel filtering techniques to support *relational learning* and infused FPs into training sets, suggesting that this practice could be used to inform other ML approaches. Third, I showed

that labeling examples as `positive' based on having multiple occurrences of a diagnosis can potentially reduce the amount of expert time needed to create training sets for phenotyping. Finally, the human interpretable phenotyping rules created from ILP could conceivably identify important clinical attributes missed by other methods, to refine phenotyping models.

5.6 CONTRIBUTORSHIP

Peggy Peissig had full access to the data in the study and takes responsibility for the integrity of the data and the accuracy of the data analysis. Peissig, Vitor Santos Costa, Michael Caldwell and David Page contributed to the conception and design of this study. Peissig and Carla Rottscheit provided data acquisition and Peissig, Costa, Caldwell, Berg Mendonca and Page analyzed and interpreted the data. Statistical analysis was done by Peissig, Costa, Berg and Page. All authors contributed to the manuscript writing and provided approval before submission to the Journal of American Medical Informatics Association in June 2013.

CHAPTER 6

Conclusion

6.1 SUMMARY

This dissertation demonstrates the innovative use of computational methods to accurately characterize research subjects using the electronic health record. It also provides anecdotal evidence that computational methods provide efficiency when phenotyping. I begin by providing background information on electronic health records, phenotyping and computational methods involving machine learning. Table 6.1 provides an overview of three studies described in Chapters 3-5. The overview includes a description of computational methods used, the accuracy of the methods and novel contributions from each study.

Chapter 3 discusses the use of a multi-modal strategy to increase the number of patients identified with cataracts for the conduct of a genome-wide association study (GWAS). This research employed 3 approaches to extract cataract status and cataract subtype information. Structured query data mining using ICD-9 and CPT codes accurately identified patients with cataracts. Cataract subtypes were not routinely coded using ICD-9 codes so computational methods were employed to extract and translate information on cataract subtypes, size and location from clinical narratives and hand written ophthalmology images. Using diagnostic codes alone only identified a small percentage of cataract subtypes. Using computational methods such as natural language processing (NLP) and optical character recognition (OCR) significantly increased the yield of subtype, size and location information that was needed to conduct a

genome-wide association study. Using only a single method (or combination of two methods) would not have provided sufficient information to conduct the GWAS study. To my knowledge the combination of these three methods has never been done when conducting a phenotyping investigation. Developing algorithms that can electronically identify and encode the cataract subtype information is thought to provide cost and time efficiencies when conducting large studies.

Table 6.1 Overview of research representing computational methods applied to EHR-driven Phenotyping

Study	Computational methods and techniques	Phenotyping accuracy	Contribution
Importance of Multi-Modal Approaches to effectively identify cataract Cases from Electronic Health Records	Combine 3 approaches for phenotyping. • Structured query data mining • Natural language processing (NLP) • Optical character recognition (OCR)	• Cataract subtype yield increased using multi-mode approach with small tradeoff to accuracy • High (> 95%) positive predictive value for cataract and subtype algorithms	• Combination of methods increased yield and provided high accuracy for cataract subtype identification
Identifying Adverse Drug Events by Relational Learning	• Inductive logic programming (ILP) • Reverse machine learning using (ILP) • Left censoring based on the initial medication data	• Identified adverse drug events for Cox2 inhibitors and Plavix	• Reverse learning method with temporal filtering methods to identify subgroups of patients (efficiency). • Censoring by matching • Temporal difference filtering
Relational Machine Learning for Electronic Health Record-Driven Phenotyping	• Inductive logic programming w/ novel • Addition of False positives • J48 & PART	• Ten phenotypes used and overall relational learning methods proved significantly better than J48 and PART for phenotyping • Human interpretable rules identify hidden attributes for phenotyping	• Relational learning for phenotyping • Automatic creation of training sets • Use of false positives to improve accuracy

Chapter 4 introduces relational machine learning along with a handful of complimentary methods to detect patients with an adverse drug event (ADE). When trying to detect ADEs, one usually does not know in advance what the ADE is. My research developed an approach that used machine learning in a manner that might seem "in reverse" and applied it to a situation where the class variable of interest (in our example ADE) was not yet known. I demonstrated this approach using Cox2 inhibitors (a medication for arthritis). In this experiment, I found that myocardial infarction (a known ADE for Cox2 inhibitors) was identified in a rule describing patients who use Cox2 inhibitors.

Several methods were developed to refine the relational learning approach for this experiment. First, data for the learning task must be left censored. If you are using medication as the class variable (or for supervision), censoring background data based on the initial medication date is needed. This method will filter out most events that happen before medication administrated and thus improve model accuracy. Second, use a scoring function that emphasizes events that take place after a censoring date. This strategy helped to identify the effects of an event (in this case and ADE for a specific medication use). Third, vetting the findings with a clinical expert is needed because of the nuances associated with EHR data. This work provides anecdotal evidence of how computational methods can be used to efficiently identify ADEs.

Chapter 5 demonstrates the novel application of relational learning to conduct EHR-driven phenotyping. Building on the ideas presented in Chapter 4, I found that using relational learning for EHR-driven phenotyping was similar in method to ADE detection. The task for phenotyping is to identify a subgroup of patients that have similar characteristics denoting a

phenotype where phenotype is the class variable. This study boasts of several contributions beginning with the use of relational learning and specifically inductive logic programming as a possible alternative method for phenotyping. We compared the performance of ILP against 2 popular decision tree methods, using area under the receiver operator characteristics (AUROC) curve, for 9 phenotype models. The results from a binomial classification two-sided significance test indicated a significant difference favoring ILP + false positives when compared to J49 ($p=0.003$) and PART ($p=0.039$). There was no significant difference when testing accuracy and F-Measure. The second contribution highlights the automatic identification of training subjects using a high frequency of phenotype related ICD-9 codes for positive examples and using subjects with no ICD-9 codes for negative examples. This reduced the need for clinical expert time during the phenotyping process. The next contribution dealt with refining the ILP approach by infusing patients with only a single diagnosis into the negative examples, to improve model accuracy and produce more specific rules. The resultant rules assist the phenotyping team in identifying hidden features that may contribute to the phenotyping model accuracy or saving time when building the model.

The common theme throughout these chapters has been using computational methods to improve the accuracy and/or find ways to efficiently characterizing subjects (patients), based on EHR data. This research supports my thesis statement as well as contributing novel methods to the domains of EHR-driven phenotyping and machine learning.

6.2 FUTURE WORK

This research can be extended to encompass several areas including methodology advances and new applications. Below are a few ideas for future research.

Rule-of-N. A basic practice when using structured query data mining is to use the rule-of-*n* where *n* is some number. This rule states that a patient must have *n*+ diagnoses spanning multiple days before being selected. This rule is used to filter out rule-out diagnoses, which are documented when ordering tests. This rule can also be applied to other types of clinical data including lab results and medications. From this research I found that ILP provides a series of rules that identify patients with a given phenotype. Most of the rules include a single diagnostic code (because positive example selection was based on diagnostic codes), along with one or more other features. I noticed that in some situations, ILP would learn a rule that was too general and thus, ILP invited the identification of FPs. Future research is needed to examine grouping of rules and selection of subjects based on a combination of multiple rule conditions, thereby combining the advantages of ILP and the general "rule-of-N" approach.

Expand automatic training sample selection. This research provided evidence that labeling patients having a high frequency of diagnosis codes as cases and labeling patients with no diagnosis codes as controls may be a viable option to identify training examples for future studies. Because the selected phenotyping training sets were disease-based, ICD-9 diagnosis codes were a natural fit to represent patients with a disease-based phenotype. Because not all phenotypes can be described using an ICD-9 code, future research is needed to investigate

alternatives such as using laboratory values or medications or even symptoms as selection criteria for training examples.

Cross-institution comparisons. More study is needed surrounding the portability and generalizability of relational learning algorithms (and rules) across institutions. Research addressing these two questions could be conducted: 1) Are rules created from relational learning phenotyping transportable across EHRs? and 2) Can relational learning be used in a similar fashion at other sites and produce the same results?

Insurance domain application. The relational learning phenotyping methods could be adapted and applied to the health insurance domain. Health insurers are interested in understanding the populations they insure. Could ILP be applied to insurance data to identify subgroups of members that have a high readmission rates or are high cost claimants so that insurers can proactively coordinate the member's care (or work with provider groups to coordinate care), so as to reduce health care costs?

Bibliography

Altman, D.G., and Bland, J.M. (1994). Diagnostic tests 1: sensitivity and specificity. *BMJ*, 308:1552.

Altman, D.G., and Bland, J.M. (1994 (2)). Diagnostic tests 2: predictive values. *BMJ*, 309:102.

Altman, D.G., and Bland, J.M. (1995) Absence of evidence is not evidence of absence. *BMJ*, 311:485.

Anandv V., and Downs, S.M. (2010). An empirical validation of recursive noisy OR (RNOR) rule for asthma prediction. *AMIA Annu Symp Proc, 16-20.*

Aronson, A.R. (2001). Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. *AMIA Annu Symp Proc,* 17–21.

Ashwin, S. (2001). The Aleph Manual – University of Oxford. *Aleph website*. http://www.di.ubi.pt/~jpaulo/competence/tutorials/aleph.pdf .

Barrick, M.R., Stewart, G.L., Neubert, M.J., and Mount, M.K. (1998). Relating member ability and personality to work-team process and team effectiveness. *Journal of Applied Psychology,* 83(3):377-391.

Berg, B., Peissig, P., Page, D., et al. (2010). Relational rule-learning on high-dimensional medical data. *Neural and Information Processing Systems (NIPS) Workshop on Predictive Models for Personalized Medicine*. Whistler, BC.

Bickeböller, H., Barrett, J.H., Jacobs, K.B., and Rosenberger, A. (2003). Modeling and dissection of longitudinal blood pressure and hypertension phenotypes in genetic epidemiological studies. *Genet Epidemiol*, 25:S72–7.

Botis, T., Hartvigsen, G., Chen, F., and Weng, C. (2010). Secondary use of EHR: data quality issues and informatics opportunities. *AMIA Summits Transl Sci Proc*, 2010:1-5.

Breiman, L., Friedman, J., Stone, C.J., et al. (1984). *Classification and regression trees.* Chapman & Hall/CRC, New York.

Burnside, E.S., Davis, J., Chatwal, J., Alagoz, O., Lindstrom, M.J., Geller, B.M., Littenberg, B., Kahn, C.E., Shaffer, K.A., and Page, C.D. (2009). Probabilistic computer model developed from clinical data in national mammography database format to classify mammographic findings. *Radiology,* 251:663-72.

Carroll, R.J., Eyler, A.E., and Denny, J.C. (2011). Naïve electronic health record phenotype identification for rheumatoid arthritis. *AMIA Annu Symp Proc*, 2011:189–96.

Carroll, R.J., Thompson, W.K., Eyler, A.E., Madelin, A.M., Cai, T., Aink, R.M., Pacheco, J.A., Boomershine, C.S., Lasko, T.A., and Xu, H. (2012). Portability of an algorithm to identify

rheumatoid arthritis in electronic health records. *Journal of the American Medical Informatics Association*.

Chen, E.S., Hripcsak, G., Xu, H., Markatou, M., and Friedman, C. (2008). Automated acquisition of disease drug knowledge from biomedical and clinical documents: an initial study. *Journal of the American Medical Informatics Association*, 15(1):87–98.

Congdon, N., O'Colmain, B., Klaver, C.C., et al. (2004). Causes and prevalence of visual impairment among adults in the United States. *Arch Ophthalmol*, 122:477–85.

Congdon, N., Vingerling, J.R., Klein, B.E., et al. (2004). Prevalence of cataract and pseudophakia/aphakia among adults in the United States. *Arch Ophthalmo,* 122:487-94.

Crosslin, D.R., McDavid, A., Weston, N., et al. (2012). Genetic variants associated with the white blood cell count in 13,923 subjects in the eMERGE Network. *Hum Genet,* 131:639-52.

Davis, J., Burnside, E., Dutra, I., Page, D., Ramakrishnan, R., Santos Costa, V., and Shavlik J. (2005). View learning for statistical relational learning: with an application to mammography. *IJCAI*.

Davis, J., Lantz, E., Page, D., et al. (2008). Machine learning for personalized medicine: will this drug give me a heart attack? *International Conference of Machine Learning (ICML); Workshop on Machine Learning in Health Care Applications*. Helsinki, Finland.

Davis, J., Santos Costa, V., Berg, E., et al. (2012). Demand-driven clustering in relational domains for predicting adverse drug events. *Proceedings of the International Conference on Machine Leaning (ICML).* ICML website: http://icml.cc/discuss/2012/644.html.

Denny, J.C., Smithers, J.D., Miller, R.A., et al. (2003). "Understanding" medical school curriculum content using KnowledgeMap. *J Am Med Inform Assoc*, 10:351–62.

Denny, J.C., Spickard, A., Miller, R.A., Schildcrout, J., Darbar, D., et al. (2005) Identifying UMLS concepts from ECG impressions using KnowledgeMap. *AMIA Annu Symp Proc,* 196–200.

Denny, J.C., Miller, R.A., Waitman, L.R., et al. (2009). Identifying QT prolongation from ECG impressions using a general-purpose natural language processor. *Int J Med Inf*, 78(Suppl 1):S34–42.

Denny, J.C., Ritchie, M.D., Crawford, D.C., et al. (2010). Identification of genomic predictors of atrioventricular conduction: using electronic medical records as a tool for genome science. *Circulation.* 122:2016–21.

Denny, J.C., Crawford, D.C., Ritchie, M.D., et al. (2011). Variants near FOXE1 are associated with hypothyroidism and other thyroid conditions: using electronic medical records for genome-and phenome-wide studies. *Am J Hum Genet,* 89:529-42.

Denny, J.C., (2012). Chapter 13: Mining electronic health records in the genomics era. *PL0S Comput Diol*, 8(12): e1002823.

De Raedt, L. (2008). *Logical and relational learning*. Springer-Verlag. Berlin, Heidelberg.

Dingcheng, L., Simon, G., Pathak, J., et al. (2013). Using association rule mining for phenotype extraction from electronic health records. *AMIA Annu Symp Proc, CR1:142-146.*

Dzeroski, S., Lavrac, N., (2001). *Relational Data Mining*. Springer-Verlag Berlin Heidelberg.

Efron B, Tibshrani R. *An Introduction to the Bootstrap. Monographs on Statistics and Applied Probability*. New York, NY: Chapman and Hall 1993.

Elkin, P.L., Ruggieri, A.P., Brown, S.H., Buntrock, J., Bauer, B.A., et al. (2001). A randomized controlled trial of the accuracy of clinical record retrieval using SNOMED-RT as compared with ICD9-CM. *AMIA Annu Symp Proc,* 159–163.

Elkin, PL Trusko BE, Koppel R, Speroff T, Mohrer D, Sakji S, Gurewitz I, Tuttle M, Brown SH. (2010). Secondary use of clinical data. *Stud Health Technol Inform*. 155:14-29.

Elixhauser, A., Steiner, C., Harris, D.R., Coffey, R.M. (1998). Comorbidity measures for use with administrative data. *Medical Care,* 36:8–27.

Ellwein, L.B. and Urato, C.J. (2002). Use of eye care and associated charges among the Medicare population: 1991-1998. *Arch Ophthalmol,* 120:804–11.

Ellsworth, D.L., and Manolio, T.A. (1999).The emerging importance of genetics in epidemiologic research II. Issues in Study Design and Gene Mapping. *Ann Epidemiol*, 9:75-90.

eMERGE Electronic Medical Records and Genomics (accessed 21 July 2010). (eMERGE) Network. http://www.gwas.org/.

eMERGE Funding (Accessed 10 June 2013). NIH News website. http://www.nih.gov/news/health/aug2011/nhgri-17.htm .

Friedman, C., Alderson, P.O., Austin, J., Cimino, J.J., and Johnson, S.B.. (1994). A general natural language text processor for clinical radiology. *J Am Med Inform Assoc*, 1:161–74.

Friedman, C., Johnson, S.B., Forman, B., Starren, J. (1995). Architectural requirements for a multipurpose natural language processor in the clinical environment. In: Gardner RM, ed. *Proceedings of Ninteenth SCAMC.* Philadelphia, PA:Hanley & Belfus, 238-242.

Friedman, C., Shagina, L., Lussier, Y., and Hripcsak, G. (2004). Automated encoding of clinical documents based on natural language processing. *J Am Med Inform Assoc*, 11:392–402.

Friedman, N., Geiger, D., Goldszmidt, M. (1997). Bayesian Network Classifiers. *Machine Learning*, 29:131-163.

Friedman, N., Nachman, I., De'er D. (1999). Learning Bayesian Network Structure from Massive Datasets: The "Sparse Candidate" Algorithm. *Proceedings of the Fifteenth Conference Annual Conference on Uncertainty in Artificial Intelligence*, 206-215.

Geiger, D., and Heckerman, D. (1997). A characterization of the Dirichlet distribution through global and local parameter independence. *Ann. Statist*, 25(3):1344-1369.

Getoor, L., Taskar, B. (2007). Introduction to statistical relational learning. In: *Statistical Relational Learning*. MIT Press. Cambridge, MA.

Goryachev, S., Sordo, M., Zeng, Q.T., (2006). A suite of natural language processing tools developed for the I2B2 project. *AMIA Annu Symp Proc,* 2006:931.

Govindaraju, V. (2005). Emergency medicine, disease surveillance, and informatics. *Proceedings of the 2005 National Conference on Digital Government Research,* 2005:167-168.

Gurwitz, D., and Pirmohamed, M. (2010). Pharmacogenomics: the importance of accurate phenotypes. *Pharmacogenomics*, 11:469–70.

Gurwitz, J., Field, T., Harrold, L., J, J. R., Debellis, K., Seger, A., Cadoret, C., Fish, L., Garber, L., Kelleher, M., and Bates, D. (2003). Incidence and preventability of adverse drug events among older persons in the ambulatory setting. *JAMA,* 289:1107–1116.

Herzig SJ, Howell MD, Ngo LH, Marcantonio ER. (2009). Acid-suppressive medication use and the risk for hospital-acquired pneumonia. *JAMA*, 301(20):2120-8.

Ho, M.L., Lawrence, N., van Walraven, C., et al. (2012). The accuracy of using integrated electronic health care data to identify patients with undiagnosed diabetes mellitus. *J Eval Clin Pract*, 18:606–11.

Hripcsak, G., Knirsch, C., Zhou, L., Wilcox, A., and Melton, G. (2011). Bias associated with mining electronic health records. *J Biomed Discov Collab*, 6:48-52.

Huang, Y., and McCullagh, P. (2007). Feature selection and classification model construction on type 2 diabetic patients' data. *Artif Intell Med*. 41(3): 251-262.

Inductive logic programming. (Accessed 2010). *Wikipedia* website: http://en.wikipedia.org/wiki/inductive_logic_programming.

Jeff, J.M., Ritchie, M.D., Denny, J.C., Kho, A.N., Ramirez, A.H., Crosslin, D., Armstrong, L., Basford, M.A., Wolf, W.A., Pacheco, J.A., Chisholm, R.L., Roden, D.M., Hayes, M.G., and Crawford, D.C. (2013). Generalization of variants identified by genome-wide association studies for electrocardiographic traits in African Americans. *Ann Hum Genet*. Doi:10.1111/ahg. 12023. (Epub ahead of print).

Kawaler, E., Cobian, A., Peissig, P., et al. (2012). Learning to predict post-hospitalization VTE risk from EHR data. *AMIA Annu Symp Proc,* 2012:436-45.

Kho, A.N., Pacheco, J.A., Peissig, P.L., et al. (2011). Electronic medical records for genetic research: results of the eMERGE Consortium. *Sci Transl Med,* 3:3–79.

Kho, A.N., Hayes, M.G., Rasmussen-Torvik, L., et al. (2012). Use of diverse electronic medical record systems to identify genetic risk for type 2 diabetes within a genome-wide association study. *J Am Med Inform Assoc,* 19:212–8.

Kim, D., and Yu, H. (2011). Figure text extraction in biomedical literature. *PLoS One,* 6(1):e15338.

Klosgen, W. (2002). *Handbook of Data Mining and Knowledge Discovery*, *Chapter 16.3:Sub-group Discovery.* Oxford University Press, New York.

Kohavi, R., and Provost, F. (1998). Special issue on applications and the knowledge discovery process. *Machine Learning,* 30(2/3).

Korb, K.B., and Nicholson, A.E. (2005). *Bayesian Artificial Intelligence*. Chapman & Hall/CRC. New York.

Kullo, I.J., Fan, J., Pathak, J., et al. (2010). Leveraging informatics for genetic studies: use of the electronic medical record to enable a genome-wide association study of peripheral arterial disease. *J Am Med Inform Assoc,* 7:568–74.

Kudyakov, R., Bowen, J., Ewen, E., et al. (2012). Electronic health record use to classify patients with newly diagnosed versus preexisting type 2 diabetes: infrastructure for comparative effectiveness research and population health management. *Popul Health Manag*, 15:3-11.

Lavrac, N., and Dzeroski, S. (1994). Inductive logic programming: techniques and applications.

Lazarou, J., Pomeranz, B.,and Corey, P. (1998). Incidence of adverse drug reactions in hospitalized patients: A metaanalysis of prospective studies. *JAMA*, 279:1200–1205.

LEADTOOLS®. Developer's Toolkit: ICR engine. (Accessed 30 September 2010). LEADTOOLS website. http://www.leadtools.com/sdk/ocr/icr.htm.

Li, L., Chase, H.S., Patel, C.O., Friedman, C., Weng, C. (2008). Comparing ICD9-encoded diagnoses and NLP-processed discharge summaries for clinical trials pre-screening: a case study. *AMIA Annu Symp Proc, 2008:*404–408.

Liao, K.P., Kurreeman, F., Li, G., Duclos, G., Murphy, S., et al. (2013). Associations of autoantibodies, autoimmune risk alleles, and clinical diagnoses from the electronic medical records in rheumatoid arthritis cases and non-rheumatoid arthritis controls. *Arthritis Rheum*. 65(3):571-81.

Lindberg, D.A., Humphreys, B.L., and McCray, A.T. (1993). The unified medical language system. *Methods Intern Med*, 32(4):281–291.

Linder, J.A., Ma, J., Bates, D.W., et al. (2007). Electronic health record use and the quality of ambulatory care in the United States. *Arch Intern Med,* 167:1400–5.

Liu, B., Hsu, W., Ma, Y. (1998). Integrating Classification and Association Rule Mining. *KDD-98*, New York.

Liu, J., Zhang, C., McCarty, C., et al. (2012). Graphical-model based multiple testing under dependence, with applications to genome-wide association studies. *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence (UAI)*.

Manolio, T.A. (2009). Collaborative genome-wide association studies of diverse diseases:programs of the NHGRI's office of population genomics. *Pharmacogenomics,* 10(2):235-241.

McCarty, C.A., Mukesh, B.N., Dimitrov, P.N., Taylor, H.R. (2003). The incidence and progression of cataract in the Melbourne Visual Impairment Project. *Am J Ophthalmol.* 2003, 136:10–7.

McCarty, C., Wilke, R.A., Giampietro P.F., et al. (2005). Marshfield Clinic Personalized Medicine Research Project (PMRP): design, methods and recruitment for a large population-based biobank. *Personalized Medicine,* 2:49–79.

McCarty, C.A., Peissig, P., Caldwell, M.A., Wilke, R.A. (2008). The Marshfield Clinic Personalized Medicine Research Project: 2008 scientific update and lessons learned in the first 6 years. *Personalized Medicine,* 5:529–42.

McCarty, C.A., Wilke, R.A. (2010). Biobanking and pharmacogenetics. *Pharmacogenomics,* 11:637–41.

McCarty, C.A., Chisholm, R.L., Chute, C.G., et al. (2011). The eMERGE Network: a consortium of biorepositories linked to electronic medical records data for conducting genomic studies. *BMC Med Genomics,* 4:13.

MedLEE™ Natural Language Processing. (Accessed 23 January 2011). MeLEE website: http://www.nlpapplications.com/index.html .

Melton, G.B., and Hripcsak, G. (2005). Automated detection of adverse events using natural language processing of discharge summaries. *Journal of the American Medical Informatics Association*, 12(4):448–457.

Mendonca, E.A., Haas, J., Shagina, L., Larson, E., Friedman, C. (2005). Extracting information on pneumonia in infants using natural language processing of radiology reports. *Journal of Biomedical Informatics*, 38:314–21.

Milewski, R., Govindaraju, V. (2004). Handwriting analysis of pre-hospital care reports. *Proceedings of CBMS,* 2004:428-433.

Mitchell, T. (1997). *Machine Learning*. McGraw Hill, Boston, Massachusetts.

Moore, A.W. (Accessed 30 May 2003). Information gain. Information gain website. http://cs.uwindsor.ca/~angom/teaching/cs574/infogain11.pdf

Muggleton, S., King, R.D., Sternberg, M.J. (1992). Protein secondary structure prediction using logic-based machine learning. *Protein Eng,* 5:647-57.

Muggleton, S. (1995). Invers entailment and Progol. New Generation Computing, 114(1-2):283-295.

National Library of Medicine. *Fact Sheet* UMLS® Metathesaurus®. (Accessed 13 January 2003). UMLS Fact Sheet website: http://www.rsna.org/radlex/committee/UMLSMetaThesaurusFactSheet.doc .

NIH Budget Categories. (Accessed 15 April 2013). NIH Budget Category website: http://report.nih.gov/rcdc/categories/.

NIH Budget. (Accessed 15 April 2013). NIH Budget website: http://www.nih.gov/about/budget.htm.

NIH RCDC Funding. (Accessed 2 May 2013) Estimates of funding for various research condition, and disease categories (RCDC) website: http://report.nih.gov/categorical_spending.aspx .

Newton, K.M., Peissig, P.L., Kho, A.N., et al. (2013). Validation of electronic medical record-based phenotyping algorithms: results and lessons learned from the eMERGE network. *Journal of the American Medical Informatics Association*, 20:e147–54.

Page, D., and Srinivasan, A. 2003. Ilp: A short look back and a longer look forward. *Journal of Machine Learning Research*, 4:415–430.

Page, D., Santos Costa, V., Natarajan, S., et al. (2012). Identifying adverse drug events by relational learning. In: Hoffman J, Selman B, eds. *Proceedings of the 26th Annual AAAI Conference on Artificial Intelligence*. AAAI website: http://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/view/4941 .

Page, D. (Accessed 10 June 2013). CS 760: Machine Learning website. http://pages.cs.wisc.edu/~dpage/cs760/ .

Pakhomov, S. and Weston, S.A. (2007). Electronic medical records for clinical research: application to the identification of heart failure. *Am J Manag Care,* 13(6 Part 1): 281-288.

Pakhomov, S., Hemingway, H., Weston, S.A., Jacobsen, S.J., Rodeheffer, R., and Roger, V.L. (2007). Epidemiology of angina pectoris: role of natural language processing of the medical record. *Am Heart J*, 153(4): 666-673.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligence Systems*. Morgan Kaufmann, San Matco, CA.

Peissig, P., Sirohi, E., Berg, R.L., et al. (2006). Construction of atorvastatin dose-response relationships using data from a large population-based DNA biobank. *Basic Clin Pharmacol Toxicol.* 100:286–88.

Peissig, P.L., Rasmussen, L.V., Berg, R.L., Linneman, J.G., McCarty, C.A., et al. (2012). Importance of multi-modal approaches to effectively identify cataract cases from electronic health records. *Journal of the American Medical Informatics Association*, 19:225–234.

Penz, J.F., Wilcox, A.B., and Hurdle, J.F. (2007). Automated identification of adverse events related to central venous catheters. *Journal of Biomedical Informatics,* 40:174-82.

Piasecki, M. and Broda, B. (2007). Correction of medical handwriting OCR based on semantic similarity. *Intelligent Data Engineering and Automated Learning – IDEAL,* 4881:437–46.

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco.

Quinlan, J.R. (1986). Induction of decision trees. *Machine Learning*, 81-106. Springer-Verlag, Berlin/Heidelberg.

Quinlan, J.R., Cameron-jones, R.M. (1995). Induction of Logic Programs: FOIL and Related Systems. *New Generation Computing*, 13:287-312.

Rasmussen-Torvik, L.J,, Pacheco, J.A., Wilke, R.A., et al. (2012). High density GWAS for LDL cholesterol in African Americans using electronic medical records reveals a strong protective variant in APOE. *Clinical Translational Science,* 5:394-9.

Rasmussen, L.V., Peissig, P.L., McCarty, C.A., Starren, J.B. (2011). Development of an optical character recognition pipeline for handwritten form fields from an electronic health record. *Journal of the American Medical Informatics Association*, Sep 2. Epub.

Rice, J.P., Saccone, N.L., and Rasmussen, E. (2001). Definition of the phenotype. *Adv Genet*, 42:69–76.

Ritchie, M.D., Denny, J.C., Zuvich, R.L., et al. (2013). Genome- and phenome-wide analysis of cardiac conduction identifies markers of arrhythmia risk. *Circulation,* 127(13):1377-85.

Ritchie, M.D., Denny, J.C., Crawford, D.C., et al. (2010). Robust replication of genotype-phenotype associations across multiple diseases in an electronic medical record. *Am J Hum Genet,* 86:560–72.

Roberts, A., Gaizauskas, R., Hepple, M., and Guo, Y. (2008). Mining clinical relationships from patient narratives. *BMC Bioinformatics*. 9(Suppl 11):S3.

Rokach, L., Maimon, O. (2010). *Data Mining and Knowledge Discovery Handbook, Chapter 9 Data Mining.* Springer, New York.

Russell, S.J., Norvig, P. (2003). *Artificial Intelligence: A Modern Approach (2nd ed.)*. Prentice Hall, New Jersey.

Sagen, C. (Accessed 1 June 2013). Absence of evidence is not evidence of absence website: http://c2.com/cgi/wiki?AbsenceOfEvidenceIsNotEvidenceOfAbsence.

Samuels, D.C., Burn, D.J., and Chinnery, P.F. (2009). Detecting new neurodegenerative disease genes: does phenotype accuracy limit the horizon? *Trends Genet,* 25(11):486-488

Saria, S., McElvain, G., Rajani, A.K., et al. (2010). Combining structured and free-text data for automatic coding of patient outcomes. *AMIA Annu Symp Proc,* 2010:712–6.

Savova, G.K., Fan, J., Ye, Z., et al. (2010). Discovering peripheral arterial disease cases from radiology notes using natural language processing. *AMIA Annu Symp Proc*, 2010:722–26.

Savova, G.K., Masanz, J.J., Ogren, P.V., et al. (2010). Mayo clinical text analysis and knowledge extraction system (cTAKES): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17:507–13.

Schulze, T.G. and McMahon, F.J. (2004). Defining the phenotype in human genetic studies: forward genetics and reverse phenotyping. *Hum Hered,* 58:131-138.

Shannon, C.E. (1948). A Mathematical theory of communication. *Bell System Technical Journal*, 27(3): 379–423.

Starren, J., and Johnson, S.B.. (1996). Notations for high efficiency data presentation in mammography. In Cimino JJ, ed. *Proceedings of the American Medical Informatics Association Annual Fall Symposium (formerly SCAMC),* Hanley & Belfus, Philadelphia, PA, 557-561.

Starren, J., Friedman, C., and Johnson, S.B. (1995). Design and development of the columbia integrated speech interpretation system (CISIS). (abstract*) Spring Conference of the American Medical Informatics Association*.

Struyf, J., Dobrin, S., and Page, D. (2008). Combining gene expression, demographic and clinical data in modeling disease: a case study of bipolar disorder and schizophrenia. *BMC Genomics*, 9:531.

Syndrome. (Accessed 30 September 2010). http://en.wikipedia.org/wiki/Syndrome

Srinivasan, A. (2004). The Aleph Manual. *www.cs.ox.ac.uk/activities/machlearn/**Aleph**/**aleph**.html*

Tesseract-OCR. engine homepage. (Accessed 30 September 2010). *Google Code* website: http://code.google.com/p/tesseract-ocr/.

Thylefors, B., Négrel, A.D., Pararajasegaram, R., and Dadzie, K.Y. (1995). Available data on blindness (update 1994). *Ophthalmic Epidemiol,* 2:5–39.

Timofeev, R. (Accessed 15 May 2013). Classification and regression trees (CART) theory and applications. Website: http://tigger.uic.edu/~georgek/HomePage/Nonparametrics/timofeev.pdf .

Thilaganathan, B., Athanasiou, S., Ozmen, S., et al. (1994). Umbilical cord blood erythroblast count as an index of intrauterine hypoxia. *Arch Dis Child Fetal Neonatal Ed,* 70:F192–4.

Van Sickle, D., Magzamen, S., Maenner, M.J., et al. (2013). Variability in the labeling of asthma among pediatricians. *PLoS One,* 8:e62398.

Voorham, J., and Denig, P. (2007). Computerized extraction of information on the quality of diabetes care from free text in electronic patient records of general practitioners. *Journal of the American Medical Informatics Association*, 14:349–54.

Waudby, C.J., Berg, R.L., Linneman, J.G., Rasmussen, L.V., Peissig, P.L., Chen, L., McCarty, C.A. (2011). Cataract research using electronic health records. *BMC Ophthalmol,* 11:32.

WEKA. (Accessed 4 January 2013). WEKA website: http://weka.sourceforge.net/doc.dev/weka/classifiers/rules/PART.html.

Weiss, J., Natarajan, S., Peissig, P., et al. (2012). Statistical relational learning to predict primary myocardial infarction from electronic health records. *AAAI Conference on Innovative Applications in AI (IAAI)*.

Wilke, R.A., Xu, H., Denny, J.C., et al. (2011). The emerging role of electronic medical records in pharmacogenomics. *Clin Pharmacol Ther,* 89:378–86.

Woolley, A.W., Gerbasi, M.E., Chabris, C.F., Kosslyn, S.M., and Hackman, J.R. (2008). Bring in the experts how team composition and collaborative planning jointly shape analytic effectiveness. *Small Group Research*, 39(3)352-371.

Wojczynski, M.K. and Tiwari, H.K. (2008). Advances in genetics. Elsevier Inc. 60:75-105.

Wojczynski, M.K. and Tiwari, H.K. (2008). Definition of phenotype. *Adv Genet,* 60:75–105.

Wrobel, S. (1997). An algorithm for multi-relational discovery of subgroups. In *PKDD*.

Wu, J., Roy, J., and Stewart, W.F. (2010). Prediction modeling using EHR data: challenges, strategies, and a comparison of machine learning approaches. *Med Care,* 48:S106-13.

Xu, H., Fu, Z., Shah, A., et al. (2011). Extracting and integrating data from entire electronic health records for detecting colorectal cancer cases. *AMIA Annu Symp Proc,* 2011:1564-72.

Zinner, D.E. and Campbell, E.G. (2009). Life-Science research within US academic medical centers. *JAMA*, 302 (9):956-976.

Zelezn'y, F. and Lavrac, N. (2006). Propositionalization-based relational subgroup discovery with RSD. *Machine Learning*, 62(1-2):33–63.

Zhou, X.H., Obuchowski, N.A., and McClish, D.K. (2002). *Statistical Methods in Diagnostic Medicine*. John Wiley and Sons, Inc, New York, NY.

Appendix A: Multi-modal Cataract Validation Results

Table A.1: Detailed results for the cataract subtype multi-modal validation.

						Weighted Statistics based on Sampling Strata				Statistics based on Relevant Data			
	Relevant Cohort Size[4]	Manually Reviewed[4]	Electronic Subtype Yield[5]	Manual Subtype Yield[5]	True Positives	PPV[6]	NPV[6]	Sensitivity[6]	Specificity[6]	PPV	NPV	Sensitivity	Specificity
Nuclear Sclerotic[1]													
CDM on Coded Diagnosis	752	629	493	612	410	99.5%	1.9%	13.8%	96.2%	99.5%	3.8%	67.0%	80.0%
NLP	3862	2972	1213	2848	1129	99.8%	2.7%	38.1%	96.2%	99.8%	2.3%	39.6%	95.2%
NLP or Diagnosis	3869	2977	1267	2852	1165	99.8%	2.3%	40.8%	95.2%	99.8%	2.7%	39.3%	96.2%
OCR	3236	2395	813	2266	700	100.0 %	2.2%	23.6%	100.0%	100.0 %	2.4%	30.9%	100.0%
NLP or OCR	4233	3115	1806	2944	1629	99.9	3.6	55.0%	96.2%	99.9%	3.6%	55.3%	96.1%
Overall (CDM/NLP/OCR)[2] (95% CI)[3]	**4236**	**3118**	**1849**	**2946**	**1654**	**99.9**% (99.7, 100)	**3.7**% (2.7, 4.7)	**55.8**% (53.9, 57.6)	**96.2**% (9.03, 100)	**99.9**% (99.7, 100)	**3.7**% (2.6, 4.7)	**56.1**% (54.2, 58.0)	**96.1**% (90.0, 100)
Posterior Sub-capsular[1]													
CDM on Coded Diagnosis	752	629	287	287	224	97.0%	66.5 %	21.5%	99.6%	97.0%	81.5 %	78.0%	97.5%
NLP	3862	2972	404	1002	333	97.4%	69.6 %	32.0%	99.4%	97.4%		33.2%	99.4%
NLP or Diagnosis	3869	2977	463	1003	378	97.4%	71.0 %	36.3%	99.4%	97.4%	71.1 %	37.7%	99.4%
OCR	3236	2395	103	839	82	87.2%	62.8 %	7.9%	99.3%	87.2%	61.1 %	9.8%	99.0%
NLP or OCR	4233	3115	473	1035	383	94.8%	71.0 %	36.8%	98.7%	94.8%	70.9 %	37.0%	98.7%
Overall (CDM/NLP/OCR)[2] (95% CI)[3]	**4236**	**3118**	**529**	**1036**	**425**	**95.1**% (92.9, 97.1)	**72.3** % (70.6, 74.2)	**40.9**% (37.7, 43.8)	**98.6**% (98.0, 99.2)	**95.1**% (93.0, 96.8)	**72.3** % (70.4, 74.0)	**41.0**% 38.0, 43.9)	**98.6**% (98.0, 99.1)

Cortical[1]													
CDM on Coded Diagnosis	752	630	3	384	1	100.0 %	25.1 %	0.0%	100.0%	100.0 %	32.3 %	0.3%	100.0%
NLP	3862	2972	500	2015	469	97.7%	29.7 %	22.2%	98.4%	97.7%	30.3 %	23.3%	98.4%
NLP or Diagnosis	3869	2978	503	2017	470	97.7%	29.7 %	22.2%	98.4%	97.7%	30.4 %	23.3%	98.4%
OCR	3236	2396	297	1691	238	92.6%	26.8 %	11.2%	97.3%	92.6%	23.8 %	14.1%	96.0%
NLP or OCR	4233	3116	762	2101	673	95.7%	32.0 %	31.8%	95.8%	95.7%	32.0 %	32.0%	95.7%
Overall (coded/NLP/OCR)[2] (95% CI)[3]	**4236**	**3119**	**765**	**2102**	**674**	**95.7**% (94.1, 97.2)	**32.0** % (29.9, 34.1)	**31.9**% (30.0, 33.8)	**95.8**% (94.3, 97.2)	**95.7**% (94.3, 97.3)	**32.1** % (30.0, 34.01)	**32.1**% (29.9, 34.1)	**95.7**% (94.2, 97.2)

PPV= positive predictive value ; NPV= negative predictive value; CDM= conventional data mining; NLP= natural language processing; OCR= optical character recognition

[1]On 4309 cases meeting cataract phenotype definition.

[2]Weighted average using CDM, NLP, and OCR strategies to reflect the properties as would be expected for the full PMRP cohort.

[3]95% Confidence Interval (CI) based on 1000 bootstrap samples.

[4]Unique case counts

[5]Yield represents total number of cataract subtyped identified. There may be up to 2 subtypes noted for each subject, but the subject will only be counted once.

[6]Statistics are weighted based on sampling fractions to reflect the properties as would be expected for the full PMRP cohort.

Shown are the multi-modal cataract subtype validation statistics. Statistics were calculated using two different approaches: 1) weighted based on the sampling strata, to reflect the properties as would be expected for the full PMRP cohort; and 2) based on the availability of relevant data. A multi-mode cataract sub-typing strategy, consisting of conventional database mining (CDM), natural language processing (NLP) on electronic clinical narratives and optical character recognition (OCR) utilizing eye exam images was used to electronically classify cataract subtypes. The electronic subtype classifications were then compared to manually abstracted subtype classifications to determine the accuracy of the multi-modal components.

Appendix B: Detailed Methods for a Cataract Phenotype Example

Below are steps outlining our methods for this research. We demonstrate these steps using the cataract phenotype as an example. The steps in this document reference Appendix C which contains real-life examples of records formatted for this research and scripts.

Acronyms used in this description:

POS	refers to positive example subjects or cases;
NEG	refers to negative example subjects or controls;
Training	refers to subjects that were used to create the model;
Testing	refers to the manually validated "gold standard" subjects that were used to test the phenotyping models.
FP	refers to subjects that have a single diagnosis and could potentially be misclassified as a positive example;
Cat	refers to the prefix used for Cataract phenotype that is demonstrated in this example;
ALEPH	An inductive programming system that is available as open source. ALEPH stands for **A** **L**earning **E**ngine for **P**roposing **H**ypotheses (Aleph). http://www.cs.ox.ac.uk/activities/machlearn/Aleph/aleph.html#SEC1
WEKA	A collection of machine learning algorithms for solving data mining problems implemented in Java and available as open source. http://www.cs.waikato.ac.nz/ml/weka/

1) Prepare TRAINING Datasets: Positive and Negative examples

Subjects	Step	Description
Test & Training	1	Subjects for this investigation come from the Marshfield Clinic research data warehouse. Prior to identifying POS, NEG and FPs as training examples (per each phenotype), testing subjects were removed from the sampling frame.
Training	2	Overview: Theoretically, a patient with many diagnoses of a condition is more likely to have the condition when compared to a patient that has only a single coded diagnosis. Based on that premise, we counted the number of diagnoses for each patient (one unique diagnosis per day) and used the frequency count to determine POS, NEG and FPs subject classifications. An example was labeled as POS if it had several diagnoses (frequency count between 15-20+ that spanned multiple days). Refer to Table 1 for the exact number of diagnoses used for each phenotype. A pool of potential POS was constructed before randomly selecting a subset of POS for model building (referred to as the training set). We selected a random sample because we wanted to reduce the size of the background data files for each analysis to a manageable size. For each randomly selected POS in our training set, we randomly selected a NEG (ICD-9 code of phenotype not present in patient's medical history) from a pool of similar age and gender patients (Figure 2 provides overview of sampling). Patients having only a single diagnosis or several diagnoses on a single day were not used as POS, because we were unsure if the diagnosis was due to a coding error or to a rule-out-diagnosis (required for ordering medical, radiological, or

	laboratory tests). We refer to these patients as false positives (FPs) and used them to refine our relational learning modeling approach.
3	Identify patients with ICD-9 Codes for the phenotype of interest (refer to Table 1-for ICD-9 codes)
	a) Select training subjects using the phenotype ICD-9 codes
	b) Determine frequency of ICD-9 codes for each subject
	c) Select subjects with frequency >= (Minimum # ICD-9 Codes Used… Table 1) and label subject as POS
	d) Select subjects with diagnosis code frequency = 1 and label subject as FP
	e) Select subjects with no diagnosis codes and label as NEG
4	If the number of Positives is > Table 1-"# Training Positives" then: Randomly sample Positives – select the a sample equal to # of Positive (found in Table 1)
5	Match NEG to POS
	a) Create "match" groups of possible NEGs for each POS subject (by gender and age (within 5 years) .
	b) Randomly select a NEG subject for each POS using the sample groups. Once a NEG subject is matched it cannot be reused.
6	Randomly select a sample of FPs (to test the sensitivity of adding FPs to the ILP scoring function; a) 4 evaluations (models were developed and true positive, false positive, true negative and false negative were calculated and statistics calculated from those values) were done using contributions of 25%, 50% 75% or 100% of the number of POS subjects). b) FP subjects were added to the scoring function files in the following step.
7	Construct scoring function files: a) Constructing ILP scoring function training files: a. train.f file representing POS examples: i. use only POS(*after*) examples ii. format record as: cat('111aaa222'). (where: cat=cataract and predicate of the record; and 111aaa222 = patient identifier;) iii. combine POS example records into a single file and label file as: train.f b. train.n file representing NEG examples: i. combine NEG(*after*) and POS(*before*) into a single file and label train.n ii. format *before* record by inserting a "b" in front of the patient identifier. The "b" corresponds to the *before* censored records used as background data. *After* records do not require a prefix. 1. Example of *before* record: cat('b222aaa222'). 2. Example of *after* record: cat('222aaa222'). b) Constructing ILP+FP scoring function training files: a. train.f file representing POS examples: i. use only POS(*after*) examples ii. format record as: cat('111aaa222').

(where cat=cataract and predicate of the record;
and 111aaa222 = patient identifier;)

iii. combine POS example records into a single file and label file as: train.f

b. train.n file representing NEG examples:

i. combine NEG(*after*) and POS(*before*) and False Positive records -FP(*after*) into a single file and label train.n

ii. format *before* records by inserting a "b" in front of the patient identifier. The "b" corresponds to the *before* censored records used as background data. *After* records do not require a prefix.

1. Example of *before* record:
cat('b222aaa222').
2. Example of *after* record:
cat('222aaa222').

2) Prepare TESTING Datasets: Case (Positive) and Control (Negative) for ALEPH

Subjects	Step	Description
Test	8	Format POS records using the following format and then combine all records into a single file and label test.f cat('333aaa222').
	9	Format negative example records using the following format and then combine all records into a single file and label test.n cat('444aaa222').

3) Prepare BACKGROUND Data for ALEPH

Subjects	Step	Description
Training		Dates were converted in the data sources to age_at_event. In the following overview we use the term date but in reality it was calculated as age_at_event = (event date – date of birth). The age_at_event variable replaced all dates appearing on data source records to anonymize the records that were used in this investigation. A "censor date" was determined for each POS and FPs patient , as 30 days prior to the initial or incident diagnosis event. Records with medical events occurring less than 5 years prior to the "censor date" were labeled as *Before* data. *Before* records were labeled by placing the letter "b" in front of the patient identifier. Records with medical events happening up to 5 years after the censor date were labeled as After data (Figure 3) and no prefix was added to the patient identifier. Records for each NEG were similarly divided but based on the censor date of the corresponding POS (since NEGS have no incident diagnosis date). All records from the background knowledge data source were evaluated based on comparing the event date to the censor date.
	10	Using POS from Step 3b, find the earliest phenotype diagnosis date
	11	Assign earliest diagnosis date (Step 10) to matched NEG (Step 5)
	12	Find earliest phenotype diagnosis date for FP (Step 3c)
	13	Using earliest diagnosis date (Step 10-12), filter ALL electronic health records based on the following descriptions. (Do for each data source: diagnoses, laboratory results, medications, procedures, symptoms) a) Create *after* records:

		i) Include only records 30 days before or 5 years after the earliest diagnosis date
		ii) Format records for ALEPH
		b) Create *before* records:
		i) Include only records 5 years to 30 days before the earliest diagnosis date
		ii) Place a "b" in front of the Patient_id on all records (indicating a *before* record)
		iii) Format records for ALEPH – See Appendix C for examples of background data source records (examples 1-7)
Training	14	Combine *Before* and *After* records into a single file (Do for each data source)
Test	15	Format Test EHR records (for each data source) without adding the prefix of "b"; See examples for each data source in Appendix C.
Test and Training	16	Combine Training and Testing EHR records into a single file (for each data source)
4) Copy scripts to ALEPH environment and run the runALEPH.sh script (refer to Appendix C-Section ?)		
Training & Test	17	Check setting in ALEPH .b parameter file (Appendix C-8). In this example the file is named cat.b Make sure that files are named as follows: Training Pos file -> cataract.f Training Neg file -> cataract.n Testing Pos file -> cataract.testf Testing Neg file -> cataract.testn Diagnosis file -> diagnosis.pro Lab results file -> lab.pro Medications file -> drug.pro Symptoms file -> symptom.pro Procedures file -> procedure.pro Gender file -> gender.pro Vitals file -> vitals.pro
	18	Run the script runAleph.sh found in Appendix C-9. Prompt> ./runAleph cat nohup & The script will run in the background and place rules and relevant information in a file named: Logcat_output
5) Create ILP Rule files for Area Under Receivers Operator Characteristics curve (AUROC) using WEKA Bayes Net Classification		
Training & Test	19	Copy the log file in Step 18, to another file named: theory0.yap and edit it by removing everything except the rules located at the end of the log file. An example of a rule is: cat(A) :- diagnoses(A,B,C,'366.10','Senile Cataract Nos',D).
	20	Copy or move cataract.f to train.f Copy or move cataract.n to train.n Copy or move cataract.testf to test.f Copy or move cataract.testn to test.n
	21	Copy the script gen_bmap into the directory and execute the following command to

		create a binomial file that can be used in WEKA. Gen_bmap is found in Appendix C-11; Prompt> xyap –L gen_bmap – 0 cat.b
	22	Several files will be created from Step 21. You will need to transfer the .arff files to WEKA for analysis. These files have combined the data definition with the data.
6) Run WEKA		
Training & Test	23	Load .arff data files that were created in Step 22 into WEKA.
	24	Run BayesNet (estimator=SimpleEstimator – A0.5, searchAlgorithm=Tan-SBAYES)

7) Prepare files and run WEKA PART and J48 Classifiers		
Subjects	**Step**	**Description**
Training		The same training and validation sets used in relational learning model building, are also used for the classical ML approaches. EMR facts from diagnoses, laboratory, medications, procedures and symptoms were used for this analysis. We limited features to unique facts found in the data sources. Features are selected if they share by more than 0.25% of the training subjects. Frequency counts for each unique feature, by data source are combined into a single record for each patient along with a class variable indicating if the patient was a POS or NEG example. All training set patient records are combined into a single "training" file for analysis. A testing file is developed using validation sample subjects. The same features identified by the training set are used for the validation samples when constructing unique patient feature records.
	25	For each data source: Count the frequency of each unique codes (features) represented in data source for each patient (Code_frequency_by_patient). (Data source = diagnoses, procedures, lab results, medications, symptoms)
	26	Select analysis features for each data source: a) Features must be used by 0.25% of subjects; Multiply # in training subjects by 0.0025% to get cut-off point (cut-off). b) For each feature ,count the number of patients that have the feature (Patients_per_Feature). c) If Patients_per_Feature >= cut-off then use the feature for model building;
	27	Create 1 analysis record per patient; a) Limit features to features identified in Step 5c and record the Code_frequency_by_patient (Step 1) b) Combine all features (Step 3a) into one record per patient. c) Format each record for WEKA.
Test	28	For each data source: Count the frequency of each unique codes (features) represented in data source for each patient (Code_frequency_by_patient).
Test	29	Create 1 analysis record per patient; a) Limit features to features identified in Step 5c and record the Code_frequency_by_patient (Step 1) b) Combine all features (Step 5a) into one record per patient. c) Format each record for WEKA in .arff format
Test & Train	30	Use WEKA J48 is the Java implementation of C4.5 decision tree [1] in WEKA [2]. One of the best known and widely used decision tree algorithms, C4.5 is an improved version of Quinlan's ID3 algorithm. The C4.5 algorithm handles continuous data and can deal

sensibly with missing values by treating them as distributions of values instead of a separate value which is used in ID3. It also has the capability to develop rules by greedily pruning conditions from each rule if the estimator error is reduced, which tends to minimize over fitting of the training data. A limitation of the decision tree is that irrelevant attributes may affect the construction of a decision tree such as dates.

PART is the Java implementation of Classical And Regression Tree (CART) [3] rule-based classifier in WEKA [4]. PART is a non-parametric decision tree learning technique that is used to create decision lists (sets of rules) (Witten et al, ref needed). It uses a dependent variable that is either categorical or numeric, to create a classification or regression tree, respectively. The algorithm will identify the most significant variables by forming a partial decision tree. With each iteration it turns the "best" leaf into a rule. This rule based algorithm can handle outliers by segregating them in a separate node.

a) Run the PART classifier with the following parameters: (BinarySplits (False), confidenceFactor (0.25), minNumObj (2), numFolds (3), reduceErrorPruning (False), seed (2), unpruned (False))

b) Run the Tree - J48 classifier with the following parameters: (BinarySplits (False), confidenceFactor (0.25), minNumObj (2), numFolds (3), reduceErrorPruning (False), subTreeRaising (True), seed (1), unpruned (False))

Appendix C: File layouts and Scripts used in ILP and ILP+FP Analysis

Below are formatted record layouts and scripts that were used in our analysis. The record layouts have not been modified except for the replacement of the patient_id with an alphanumeric label; The examples come from the Cataract Phenotyping effort.

1) Background diagnosis.pro file layout:

Record description:

diagnoses('patient_id',(age_at_event*1000),'3-digit ICD-9Code','ICD-9Code,'ICD9Description','ICD9 Subcategory').

Example records:

POSITIVE subject record eample:

diagnoses('111aaa222',67304,'362','362.51','Nonexudat Macular Degen','360 - 379').

NEGATIVE subject record example with "AFTER" data:

diagnoses('222aaa222',67304,'362','362.51','Nonexudat Macular Degen','360 - 379').

NEGATIVE subject record example with "BEFORE" data:

diagnoses('b222aaa222',67304,'362','362.51','Nonexudat Macular Degen','360 - 379').

2) Background lab.pro file layout:

Record description:

lab('Patient_id', age_at_event*1000,lab_component_id,'lab result description','result','Interpretation').

POSITIVE subject record eample:

lab('111aaa222',67624,23967,'Non-HDL Cholesterol','153','Normal').

NEGATIVE subject record example with "AFTER" data:

lab('222aaa222',67624,23967,'Non-HDL Cholesterol','153','Normal').

NEGATIVE subject record example with "BEFORE" data:

lab('b222aaa222',67624,23967,'Non-HDL Cholesterol','153','Normal').

3) Background drug.pro file layout:

Record description:
hasdrug('Patient_id',age_at_event*1000,'Brand name for medication','Generic Name').

POSITIVE subject record eample:
hasdrug('111aaa222',67304,'TylenolArthritis','ACETAMINOPHEN').

NEGATIVE subject record example with “AFTER” data:
hasdrug('222aaa222',67304,'TylenolArthritis','ACETAMINOPHEN').

NEGATIVE subject record example with “BEFORE” data:
hasdrug('b222aaa222',67304,'TylenolArthritis','ACETAMINOPHEN').

4) Background procedure.pro file layout:

Record description:
has_tx('Patient_id', age_at_event*1000,'CPT code','CPTDescription','CPT SubCategory','CPT Category',DataSource,Inpatient_Outpatient).

POSITIVE subject record eample:
has_tx('111aaa222',67304,'99214','Office Outpatient Visit 25 Minutes','Office or Other Outpatient Services','Evaluation and Management',1,0).

NEGATIVE subject record example with “AFTER” data:
has_tx('222aaa222',67304,'99214','Office Outpatient Visit 25 Minutes','Office or Other Outpatient Services','Evaluation and Management',1,0).

NEGATIVE subject record example with “BEFORE” data:
has_tx('b222aaa222',67304,'99214','Office Outpatient Visit 25 Minutes','Office or Other Outpatient Services','Evaluation and Management',1,0).

5) Background gender.pro file layout:

Record description:
gender('Patient_id','gender').

POSITIVE subject record eample:
gender('111aaa222','F').

NEGATIVE subject record example with “AFTER” data:
gender('222aaa222','M').

NEGATIVE subject record example with “BEFORE” data:
gender('b222aaa222','M').

6) Background symptom.pro file layout:

Record description:

symptom('Patient_id',age_at_event*1000,'Concept Unique IdentifierI','Symptom description').

POSITIVE subject record eample:

symptom('111aaa222',70790,'C0030193','pain').

NEGATIVE subject record example with “AFTER” data:

symptom('222aaa222',70790,'C0030193','pain').

NEGATIVE subject record example with “BEFORE” data:

symptom('b222aaa222',70790,'C0030193','pain').

7) Background vitals.pro file layout:

Record description:

vitals('Patient_Id',age_at_event*1000,Vital Description,'vital measurement').

POSITIVE subject record eample:

vitals('111aaa222',68110,'Blood Pressure Diastolic','60').

NEGATIVE subject record example with “AFTER” data:

vitals('222aaa222',68110,'Blood Pressure Diastolic','60').

NEGATIVE subject record example with “BEFORE” data:

vitals('b222aaa222',68110,'Blood Pressure Diastolic','60').

8) ALEPH .b Parameter file:

```
% Cataract induce_cover phenotyping analysis script
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DEFINE "ACCEPTABLE" RULE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
:- set(noise,1). %max # neg examples a rule covers
:- set(minpos,80).
:- set(minacc,0.8). %min precision of the rule
:- set(clauselength,4).
:- set(caching,false).
:- set(i,3).
:- set(record,true).
:- set(train_pos,'cataract.f').
:- set(train_neg,'cataract.n').
```

```
:- set(test_pos,'cataract.testf').
:- set(test_neg,'cataract.testn').
%%%%%%%%%%%%%%%%%%%%%%%%%%
% SEARCH SETTINGS
%%%%%%%%%%%%%%%%%%%%%%%%%%
:- set(nodes,1000000).
%%%%%%%%%%%%%%%%%%%%%%%%%%
% MODES
%%%%%%%%%%%%%%%%%%%%%%%%%%
:- modeh(1,cat(+patient)).
% STATIC FEATURES
:- modeb(*,diagnoses(+patient,-age,-threecode,#dx_code,#dx_mc_desc,-dx_subcat)).
:- modeb(*,has_tx(+patient,-age,#cpt_code,#cpt_desc,-cpt_subcat,-cpt_cat,-cpt_source,-
cpt_pos)).
:- modeb(*,lab(+patient,-age,#labcode,#lab_desc,-lab_value,#lab_range)).
:- modeb(*,hasdrug(+patient,-age,-cx_informal,#cx_code)).
:- modeb(*,vitals(+patient,-age,#vital_desc,#vital_value)).
:- modeb(*,symptom(+patient,-age,#cui,#cui_desc)).
:- modeb(*,g_vitals(+patient,-age,#vital_desc,#vital_value)).
:- modeb(*,l_vitals(+patient,-age,#vital_desc,#vital_value)).
:- modeb(*,g_lab(+patient,-age,#lab_code,#lab_desc,#lab_value,-lab_range)).
:- modeb(*,l_lab(+patient,-age,#lab_code,#lab_desc,#lab_value,-lab_range)).
:- modeb(*,gender(+patient,#gender)).
:- modeb(*,gte(+age,#age)).
:- modeb(*,lte(+age,#age)).
:- modeb(*,after(+age,+age)).
after(X,Y) :-
        number(X), number(Y),
        !, X >= Y.
gte(X,Y) :-
        number(X), var(Y),
        !, X = Y.
gte(X,Y) :-
        number(X), number(Y),
        !, X >= Y.
lte(X,Y) :-
        number(X), var(Y),
        !, X = Y.
lte(X,Y) :-
        number(X), number(Y),
        !, X =< Y.
%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% COST Settings
%%%%%%%%%%%%%%%%%%%%%%%%%
cost(Clause,[P,N,L],Cost):-
   N1 is N*100,
   Cost is  -((P+10)/(P+N1+20)).
cost(_,_,1).
%%%%%%%%%%%%%%%%%%%%%%%%%
% DETERMINATIONS
%%%%%%%%%%%%%%%%%%%%%%%%%
:- determination(cat/1,gender/2).
:- determination(cat/1,hasdrug/4).
:- determination(cat/1,symptom/4).
:- determination(cat/1,patient/1).
:- determination(cat/1,diagnoses/6).
:- determination(cat/1,has_tx/8).
:- determination(cat/1,vitals/4).
:- determination(cat/1,lab/6).
:- determination(cat/1,g_vitals/4).
:- determination(cat/1,l_vitals/4).
:- determination(cat/1,g_lab/6).
:- determination(cat/1,l_lab/6).
:- determination(cat/1,gte/2).
:- determination(cat/1,lte/2).
:- determination(cat/1,after/2).
%%%%%%%%%%%%%%%%%%%%%%%%%
% TYPE DEFINITIONS
%%%%%%%%%%%%%%%%%%%%%%%%%
patient(X):- atom(X).
date(X):- number(X).
age(X):- number(X).
threecode(X):- atom(X).
dx_code(X):- atom(X).
cpt_code(X):- atom(X).
cui(X):- atom(X).
cui_desc(X):- atom(X).
cpt_desc(X):- atom(X).
dx_mc_desc(X):- atom(X).
dx_subcat(X):- atom(X).
vital_value(X):- number(X) ; atom(X).
vital_desc(X):- atom(X).
labcode(X):- number(X).
lab_desc(X):- atom(X).
```

```
lab_value(X):- number(X) ; atom(X).
lab_range(X):- number(X) ; atom(X).
dx_cat(X):- atom(X).
drug(X):- atom(X).
test(X):- atom(X).
value(X):- number(X) ; atom(X).
gender(X):- atom(X).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% BACKGROUND FILES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
:- style_check(-discontiguous).
% STATIC FEATURES BACKGROUND
% Data source files
:- consult/('diagnosis.pro').     %Diagnosis
:- consult('vitals.pro').         %Vitals
:- consult('lab.pro').            %Lab results
:- consult('drug.pro').                 %Medications
:- consult('symptom.pro').              %Symptoms
:- consult('gender.pro').               %Gender
:- consult('procedure.pro').            %Procedure
% TIME-BASED FEATURES BACKGROUND
g_vitals(P,A,T,V):-
      var(V),
      vitals(P,A,T,V).
g_vitals(P,A,T,V):-
      vitals(P,A,T,V1),
      number(V), number(V1),
      (V == V1 ; V1 > V).
l_vitals(P,A,T,V):-
      var(V),
      vitals(P,A,T,V).
l_vitals(P,A,T,V):-
      vitals(P,A,T,V1),
      number(V), number(V1),
      (V == V1 ; V1 < V).
g_lab(P,A,T,D,V,R):-
         var(V),
         lab(P,A,T,D,V,R).
g_lab(P,A,T,D,V,R):-
         lab(P,A,T,D,V1,R),
         number(V), number(V1),
         (V == V1 ; V1 > V).
```

```
l_lab(P,A,T,D,V,R):-
        var(V),
        lab(P,A,T,D,V,R).
l_lab(P,A,T,D,V,R):-
        lab(P,A,T,D,V1,R),
        number(V), number(V1),
        (V == V1 ; V1 < V).
```

9) Run script (runAleph.sh):

```
TARGETCLASS=$1
ALEPH=aleph.yap
YAP='xyap' # 64 bit yap
LOGFILE=Log$Run_Output
# this starts yap ("$YAP"),
# runs the commands below in yap until the EOF signal,
# redirects output to "$LOGFILE",
# and redirects any errors also to "$LOGFILE"
"$YAP" << EOF > "$LOGFILE" 2>&1
['$ALEPH'].
read_all('$TARGETCLASS').
set(record,true).
set(verbosity,0).
induce_cover.
EOF
done
```

10) Command line syntax:

```
Prompt>./runAleph cat nohup &
```

11) gen_bmap Script

```
:- source.
:- dynamic diagnoseWithinF/3.
:- style_check(all).
:- yap_flag(unknown,error).
:- initialization(main).
:- use_module(library(lists),[member/2]).

% stubs
:- op(500,fy,#).
determination(_,_).
modeh(_,_).
```

```
modeb(_,_).
set(_,_).

main :-
unix(argv([AFold, BFile])),
atom_number(AFold, Fold),
main(Fold, BFile).

main(Fold,BFile) :-
atomic_concat(['theory',Fold,'.yap'],T),
consult(BFile),
reconsult(pos:'train.f'),
reconsult(neg:'train.n'),
reconsult(tpos:'test.f'),
reconsult(tneg:'test.n'),
once(current_predicate(pos:Na/Ar)),
dynamic(Na/Ar),
consult(T),
functor(G, Na, Ar),
(
   domain(G, Fold)
;
   doweka(G, Fold)
;
   dolightsvm(G, Fold)
      ).
main(_,_).

doweka(G, Fold) :-
atomic_concat([bmap,Fold,'.arff'],F),
open(F,write, S),
weka_format(G, S),
weka_examples(pos, 1, G, S),
weka_examples(neg, 0, G, S),
close(S),
fail.
doweka(G, Fold) :-
atomic_concat([test_bmap,Fold,'.arff'],F),
open(F,write,S),
weka_format(G, S),
weka_examples(tpos,1,G,S),
weka_examples(tneg,0,G,S),
```

```
close(S),
fail.

dolightsvm(G, Fold) :-
atomic_concat([bmap,Fold,'.lsvm'],F),
open(F,write, S),
lsvm_examples(pos, 1, G, S),
lsvm_examples(neg, 0, G, S),
close(S),
fail.
dolightsvm(G, Fold) :-
atomic_concat([test_bmap,Fold,'.lsvm'],F),
open(F,write,S),
lsvm_examples(tpos,1,G,S),
lsvm_examples(tneg,0,G,S),
close(S),
fail.

domain(G, Fold) :-
atomic_concat(bmap,Fold,F),
open(F,write,B),
examples(pos,1,G,B),
examples(neg,0,G,B),
close(B),
fail.
domain(G, Fold) :-
atomic_concat(test_bmap,Fold,F),
open(F,write,B),
examples(tpos,1,G,B),
examples(tneg,0,G,B),
close(B),
fail.

examples(Mod,S,G,B) :-
call(Mod:G),
format(B,'LABEL ~d~nDistribution~n',[S]),
output_example(G,B).
examples(_,_,_,_).

output_example(G,F) :-
start_example(sriraam),
nb_setval(rules,0),
```

```
functor(G, N, A),
functor(G0, N, A),
clause(G0, Body, Ref),
nth_instance(_,I,Ref),
G = G0,
format(F,'RULE~d 1   ',[I]),
nb_getval(rules,R),
R1 is R+1,
nb_setval(rules,R1),
output_body(Body, F, sriraam),
format(F, '~n', []),
fail.

complete_example(_, _, sriraam).
complete_example(_, _, lsvm).
complete_example(F, 0, weka) :-
format(F,'0~n',[]).
complete_example(F, 1, weka) :-
format(F,'1~n',[]).

lsvm_examples(Mod,S,G,B) :-
call(Mod:G),
(
   S = 0
->
   format(B,"-1",[])
;
   format(B,"+1",[])
),
output_lsvm_example(G,B).
lsvm_examples(_,_,_,_).

output_lsvm_example(G,F) :-
start_example(lsvm),
nb_setval(rules,0),
functor(G, N, A),
functor(G0, N, A),
clause(G0, Body),
G = G0,
nb_getval(rules,R),
R1 is R+1,
nb_setval(rules,R1),
```

```
output_body(Body, F, lsvm),
fail.
output_lsvm_example(_,F) :-
nl(F),
fail.

weka_format(G, F) :-
nb_setval(rules,0),
format(F,'@RELATION bmap~n~n',[]),
nb_setval(attributes,0),
clause(G, Body),
has_goal(Body,B),
nb_getval(rules,R0),
R is R0+1,
numbervars(B,1,_),
format(F,'% ~w~n@ATTRIBUTE a~d {1,0}~n',[B,R]),
nb_setval(rules,R),
fail.
weka_format(_, F) :-
format(F,'@ATTRIBUTE class {1,0}~n~n@DATA~n',[]).

weka_examples(Mod,S,G,B) :-
call(Mod:G),
output_weka_example(G,S,B).
weka_examples(_,_,_,_).

output_weka_example(G,_,F) :-
start_example(weka),
nb_setval(rules,0),
functor(G, N, A),
functor(G0, N, A),
clause(G0, Body),
G = G0,
nb_getval(rules,R),
R1 is R+1,
nb_setval(rules,R1),
output_body(Body, F, weka),
fail.
output_weka_example(_,S,F) :-
complete_example(F, S, weka),
fail.
```

```
has_goal((B1,B2), B) :- !,
(
   has_goal(B1, B)
;
   has_goal(B2, B)
).
has_goal(B, B).


output_body((B1,B2), F, Classifier) :- !,
output_body(B1, F, Classifier),
output_body(B2, F, Classifier).
output_body(G, F, Classifier) :- !,
(
  catch(call(G),_,fail)
->
  output_bit(pos, Classifier, F)
       ;
  output_bit(neg, Classifier, F)
       ).

output_bit(pos, weka, F) :-
  format(F, "1,", []).
output_bit(neg, weka, F) :-
  format(F, "0,", []).
output_bit(pos, lsvm, F) :-
  nb_getval(atts, A0),
  A1 is A0+1,
  nb_setval(atts, A1),
  format(F, " ~d:1", [A1]).
output_bit(neg, lsvm, _) :-
  nb_getval(atts, A0),
  A1 is A0+1,
  nb_setval(atts, A1).
output_bit(pos, sriraam, F) :-
  format(F, " 1", []).
output_bit(neg, sriraam, F) :-
  format(F, " 0", []).
start_example(weka).
start_example(sriraam).
start_example(lsvm) :-
nb_setval(atts,0).
```

Appendix D: Inductive Logic Programming Rules – Cataract Phenotype

Below are inductive logic programming rules (ILP) that were produced when running the ILP only and ILP + False Postives (FP) methods for the Cataract Phenotyping. ILP provides these rules and the number of subjects that were identified as positive (POS) and negative (NEG). We did not publish the number of POS and NEG subjects that qualified for each rule due to differential privacy concerns. However we can say that a minimum of 80 POS subjects had to be identified in order for it to be considered.

1) **ILP only rules for the Cataract Phenotype**

```
cat(A) :-
  diagnoses(A,B,C,'366.10','Senile Cataract Nos',D).

cat(A) :-
  diagnoses(A,B,C,'366.9','Cataract Nos',D).

cat(A) :-
  diagnoses(A,B,C,'366.16','Senile Nuclear Cataract',D).

cat(A) :-
  diagnoses(A,B,C,'366.12','Incipient Cataract',D).

cat(A) :-
  has_tx(A,B,'92012','Ophth Medical Xm&Eval Intermediate Estab Pt',C,D,E,F),
gte(B,72753), has_tx(A,G,'80007','7 Clinical Chemistry Tests',H,I,E,F).

cat(A) :-
  has_tx(A,B,'92014','Ophth Medical Xm&Eval Comprhnsv Estab Pt 1/> Vst',C,D,E,F),
lab(A,G,20282,'Glucose',H,'Normal'), gte(G,78030).

cat(A) :-
  has_tx(A,B,'92015','Determination Refractive State',C,D,E,F),
has_tx(A,G,'99215','Office Outpatient Visit 40 Minutes',H,I,E,F), gte(G,77884).
```

2) **ILP+FP rules for the Cataract Phenotype**

```
cat(A) :-
```

```
  diagnoses(A,B,C,'366.10','Senile Cataract Nos',D), diagnoses(A,E,C,'366.9','Cataract
Nos',D).

cat(A) :-
  diagnoses(A,B,C,'715.90','Degenerative Joint Disease',D),
diagnoses(A,E,F,'366.10','Senile Cataract Nos',G), after(E,B).

cat(A) :-
  diagnoses(A,B,C,'366.10','Senile Cataract Nos',D), lab(A,E,20234,'Thyroxine
(T4)',F,'Normal'), after(B,E).

cat(A) :-
  diagnoses(A,B,C,'366.9','Cataract Nos',D), diagnoses(A,E,F,'*NA','*Not Available',G),
lab(A,E,20723,'Cholesterol',H,'High').

cat(A) :-
  diagnoses(A,B,C,'366.10','Senile Cataract Nos',D), has_tx(A,E,'99024','Postop Follow
Up Visit Related To Original Px',F,G,H,I), after(B,E).

cat(A) :-
  diagnoses(A,B,C,'366.16','Senile Nuclear Cataract',D), lab(A,E,20181,'Hematocrit
(Hct)',F,'Normal'), after(B,E).

cat(A) :-
  diagnoses(A,B,C,'366.10','Senile Cataract Nos',D), diagnoses(A,E,F,'*NA','*Not
Available',G), has_tx(A,B,'92014','Ophth Medical Xm&Eval Comprhnsv Estab Pt 1/>
Vst',H,I,J,K).

cat(A) :-
  diagnoses(A,B,C,'366.16','Senile Nuclear Cataract',D),
lab(A,E,20723,'Cholesterol',F,'High'), after(B,E).

cat(A) :-
  diagnoses(A,B,C,'366.10','Senile Cataract Nos',D), has_tx(A,E,'99213','Office
Outpatient Visit 15 Minutes',F,G,H,I), has_tx(A,J,'92012','Ophth Medical Xm&Eval
Intermediate Estab Pt',K,L,H,I).

cat(A) :-
  gender(A,'F'), diagnoses(A,B,C,'366.10','Senile Cataract Nos',D),
diagnoses(A,E,F,'715.90','Degenerative Joint Disease',G).

cat(A) :-
  diagnoses(A,B,C,'455.6','Hemorrhoids Nos',D), diagnoses(A,E,F,'366.10','Senile
Cataract Nos',G), after(E,B).

cat(A) :-
```

```
  diagnoses(A,B,C,'366.10','Senile Cataract Nos',D), diagnoses(A,E,F,'354.0','Carpal
Tunnel Syndrome',G).

cat(A) :-
  diagnoses(A,B,C,'366.10','Senile Cataract Nos',D),
diagnoses(A,E,F,'278.0','Obesity',G).

cat(A) :-
  diagnoses(A,B,C,'366.9','Cataract Nos',D), diagnoses(A,E,F,'*NA','*Not Available',G),
lab(A,H,20813,'Differential Monocyte',I,'Normal').

cat(A) :-
  diagnoses(A,B,C,'366.10','Senile Cataract Nos',D), gte(B,77333),
diagnoses(A,E,F,'367.9','Refraction Disorder Nos',D).

cat(A) :-
  diagnoses(A,B,C,'780.4','Dizziness And Giddiness',D), diagnoses(A,E,F,'366.10','Senile
Cataract Nos',G), after(E,B).

cat(A) :-
  diagnoses(A,B,C,'465.9','Acute Uri Nos',D), diagnoses(A,E,F,'366.10','Senile Cataract
Nos',G), after(E,B).

cat(A) :-
  diagnoses(A,B,C,'366.10','Senile Cataract Nos',D), lab(A,E,20451,'Total
Cholesterol/HDL Ratio',F,'Normal'), after(B,E).

cat(A) :-
  diagnoses(A,B,C,'366.10','Senile Cataract Nos',D), lab(A,E,20387,'GGT',F,'Normal'),
lab(A,G,20707,'HDL Cholesterol',H,'Normal').

cat(A) :-
  diagnoses(A,B,C,'366.16','Senile Nuclear Cataract',D), has_tx(A,E,'99212','Office
Outpatient Visit 10 Minutes',F,G,H,I), has_tx(A,J,'99243','Office Consultation New/Estab
Patient 40 Min',K,G,H,I).

cat(A) :-
  diagnoses(A,B,C,'86.3','Other Local Destruc Skin',D), diagnoses(A,E,F,'366.10','Senile
Cataract Nos',G), after(E,B).

cat(A) :-
  diagnoses(A,B,C,'366.10','Senile Cataract Nos',D), has_tx(A,E,'92014','Ophth Medical
Xm&Eval Comprhnsv Estab Pt 1/> Vst',F,G,H,I), has_tx(A,J,'76092','Mammogram
Screening',K,L,H,I).

cat(A) :-
```

```
  diagnoses(A,B,C,'366.10','Senile Cataract Nos',D), diagnoses(A,E,F,'*NA','*Not
Available',G), diagnoses(A,H,I,'365.04','Ocular Hypertension',D).

cat(A) :-
  diagnoses(A,B,C,'367.9','Refraction Disorder Nos',D), diagnoses(A,B,E,'366.10','Senile
Cataract Nos',D), diagnoses(A,F,G,'*NA','*Not Available',H).

cat(A) :-
  diagnoses(A,B,C,'366.10','Senile Cataract Nos',D), has_tx(A,E,'92015','Determination
Refractive State',F,G,H,I), has_tx(A,J,'92012','Ophth Medical Xm&Eval Intermediate
Estab Pt',F,G,H,I).

cat(A) :-
  diagnoses(A,B,C,'366.10','Senile Cataract Nos',D), diagnoses(A,B,E,'379.21','Vitreous
Degeneration',D).

cat(A) :-
  diagnoses(A,B,C,'13.71','Insert Lens At Catar Ext',D).

cat(A) :-
  diagnoses(A,B,C,'366.16','Senile Nuclear Cataract',D), has_tx(A,E,'71020','Radiologic
Exam Chest 2 Views Frontal&Lateral',F,G,H,I), after(B,E).

cat(A) :-
  diagnoses(A,B,C,'366.9','Cataract Nos',D), lab(A,E,20451,'Total Cholesterol/HDL
Ratio',F,'Normal'), after(B,E).

cat(A) :-
  diagnoses(A,B,C,'366.10','Senile Cataract Nos',D), diagnoses(A,E,F,'465.9','Acute Uri
Nos',G), lab(A,H,20723,'Cholesterol',I,'High').

cat(A) :-
  diagnoses(A,B,C,'366.16','Senile Nuclear Cataract',D), diagnoses(A,E,F,'*NA','*Not
Available',G), lab(A,H,20723,'Cholesterol',I,'High').

cat(A) :-
  diagnoses(A,B,C,'366.10','Senile Cataract Nos',D), diagnoses(A,E,C,'366.12','Incipient
Cataract',D).

cat(A) :-
  diagnoses(A,B,C,'366.10','Senile Cataract Nos',D), diagnoses(A,E,F,'789.0','Abdominal
Pain',G).

cat(A) :-
```

```
  diagnoses(A,B,C,'366.10','Senile Cataract Nos',D), diagnoses(A,E,F,'*NA','*Not
Available',G), has_tx(A,H,'82947','Glucose Quantitative Blood Xcpt Reagent
Strip',I,J,K,L).

cat(A) :-
  diagnoses(A,B,C,'366.9','Cataract Nos',D), gte(B,69306), has_tx(A,E,'G0008','Admin
Influenza Virus Vac',F,G,H,I).

cat(A) :-
  has_tx(A,B,'66984','Cataract Removal Insertion Of Lens',C,D,E,F).

cat(A) :-
  diagnoses(A,B,C,'786.50','Chest Pain Nos',D), diagnoses(A,E,F,'366.10','Senile
Cataract Nos',G), after(E,B).

cat(A) :-
  diagnoses(A,B,C,'366.10','Senile Cataract Nos',D), diagnoses(A,E,F,'45.23','Flex
Fiberoptic Colonosc',G).

cat(A) :-
  diagnoses(A,B,C,'366.9','Cataract Nos',D), has_tx(A,E,'90724','Influenza
Immunization',F,G,H,I).

cat(A) :-
  symptom(A,B,'C0037088','Clini'), diagnoses(A,C,D,'366.10','Senile Cataract Nos',E),
diagnoses(A,F,G,'*NA','*Not Available',H).

cat(A) :-
  diagnoses(A,B,C,'366.9','Cataract Nos',D), gte(B,68392), has_tx(A,E,'99024','Postop
Follow Up Visit Related To Original Px',F,G,H,I).

cat(A) :-
  diagnoses(A,B,C,'366.9','Cataract Nos',D), has_tx(A,E,'99242','Office Consultation
New/Estab Patient 30 Min',F,G,H,I), after(B,E).
```