INTRO TO CLINICAL DATA STUDY GUIDE

MODULE 6 - PUTTING THE PIECES TOGETHER: ELECTRONIC PHENOTYPING

LEARNING OBJECTIVES

1. Define electronic phenotyping
2. Describe the difference between a feature and a phenotype
3. Explain the purpose of electronic phenotyping
4. Describe the two main approaches to electronic phenotyping, and their strengths and weaknesses
5. Describe imperfect labelling and how it can be used in phenotyping

ELECTRONIC PHENOTYPING

Phenotype is a certain characteristic or condition of a patient that may be present or absent. The most common example is a disease.

Electronic phenotyping is a computational procedure for determining whether a patient does have or does not have the condition of interest based on electronic medical record.

The process of electronic phenotyping is found throughout clinical research.

Electronic phenotyping is useful for:

- Using observational data for research
- Recruiting into clinical trials
- Calculating quality metrics for healthcare systems
- Finding similar patients
- Sharing definitions to facilitate cross-site research

Difficulties of Phenotyping:

1. The codes might not be accurate
2. Even if the codes are correct, they might be assigned after the patient was first known to have the condition

The input to electronic phenotyping procedure will be a timeline of the patient's features. We need to choose which data to use, and which portion of the timeline to consider. We need to distinguish a feature from a phenotype.

- A **feature** is something that is directly measured
- A **phenotype** is the result of inference applied to one or more features

An **electronic phenotype** should contain the necessary and sufficient conditions of the features that should be present or absent and their required values to determine if an exposure or outcome of interest happened to a patient. It should also contain the criteria for identifying the start and end times. Locating the start time may be straightforward, it can be more difficult to determine when a condition ends.

When specifying a phenotype, it is very important to be clear about the intended meaning of that phenotype.

Evaluating a Phenotype Definition:

- Figure out exposures and outcomes
- Decide on risk thresholds
- Estimate the effects of treatments

Electronic phenotyping: Declaring the necessary and sufficient conditions of the features that should be present or absent and their required values to determine if an exposure or outcome of interest happened

It is important to realize that the accuracy of the phenotype definition depends on the research question you are trying to answer.

The last issue to consider for evaluation is how well a phenotype definition based on data from one clinical site will work at a second clinical site, often referred to as the portability of the electronic phenotype.

TWO APPROACHES TO PHENOTYPING

Approaches to Electronic Phenotyping:

1. Rule-based Phenotyping: Using rules comprised of explicit inclusion and exclusion criteria that were constructed by experts who reach consensus on the criteria in an iterative fashion

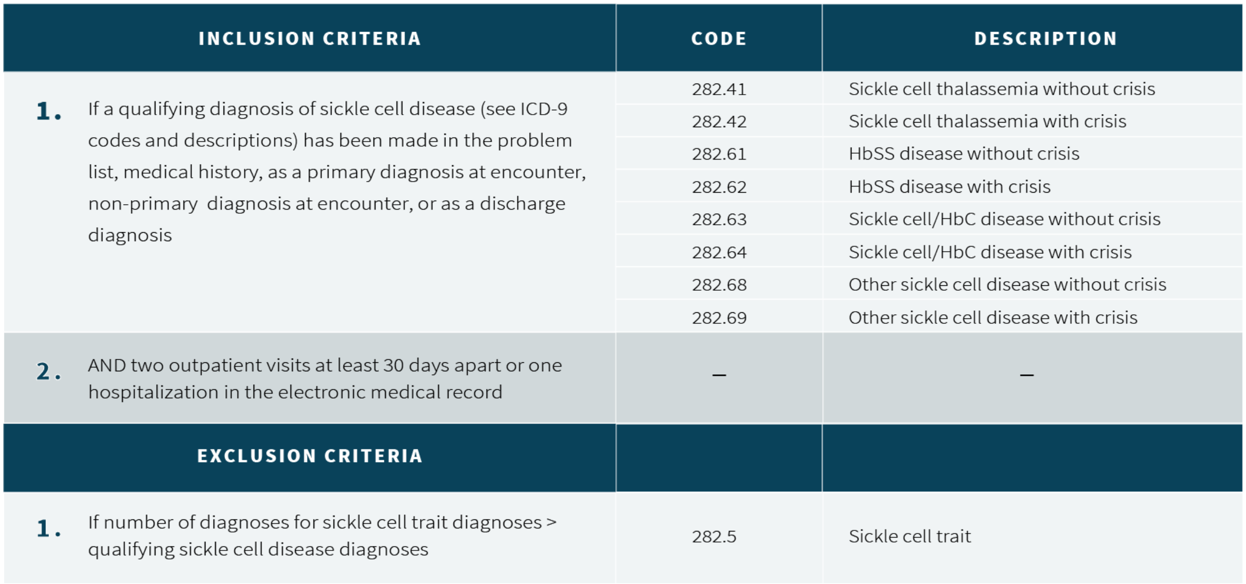
2. Probabilistic Phenotyping: Using machine learning instead of expert consensus to learn a function that assigns a probability to a patient's record for having the exposure or outcome of interest

RULE-BASED ELECTRONIC PHENOTYPING

We will use examples from the Phenotype Knowledge Base, or PheKB. This is a publicly available repository of phenotype definitions.

[**Site**: What is the Phenotype KnowledgeBase? | PheKB]

The phenotype has inclusion criteria and exclusion criteria. The inclusion criteria are features that must be present, while the exclusion criteria are features that must not be present.



INCLUSION CRITERIA	CODE	DESCRIPTION
1. If a qualifying diagnosis of sickle cell disease (see ICD-9 codes and descriptions) has been made in the problem list, medical history, as a primary diagnosis at encounter, non-primary diagnosis at encounter, or as a discharge diagnosis	282.41	Sickle cell thalassemia without crisis
	282.42	Sickle cell thalassemia with crisis
	282.61	HbSS disease without crisis
	282.62	HbSS disease with crisis
	282.63	Sickle cell/HbC disease without crisis
	282.64	Sickle cell/HbC disease with crisis
	282.68	Other sickle cell disease without crisis
	282.69	Other sickle cell disease with crisis
2. AND two outpatient visits at least 30 days apart or one hospitalization in the electronic medical record	–	–
EXCLUSION CRITERIA		
1. If number of diagnoses for sickle cell trait diagnoses > qualifying sickle cell disease diagnoses	282.5	Sickle cell trait

Example of sickle cell disease: The inclusion criteria has two parts: a set of ICD-9 codes and the requirement for one hospitalization or two clinic visits for this condition. The exclusion criteria has one part, which says that if the patient has more diagnoses for sickle cell trait than for sickle cell disease, then do not determine the patient to have sickle cell disease.

Example of type 2 diabetes mellitus: This phenotype is expressed as a flowchart, which they call an "algorithm". Starting with the records for a patient stored in the EMR; following "Yes" and "No" on whether the patient has diagnosed of Type 1 diabetes and check Type 2 diabetes diagnosis, prescription, or relevant abnormal lab value. The point here is that diagnosis codes are not accurate enough to be used on their own, but need to be augmented by prescriptions of medications, specific lab values, and consideration of the relative timing of events, stressing the need to have codified "prior knowledge" and adopting a timeline view of the patient record.

It is important to remember that each of the sources of data can be useful, that combinations of the sources are more accurate, and that the relative value of a source varies with the disease of interest.

Constructing a rule-based phenotype definition:

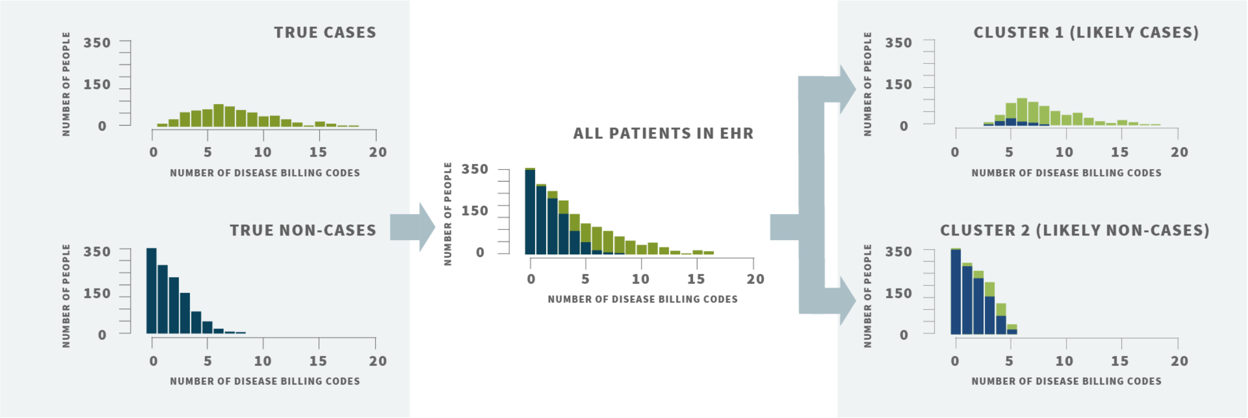
1. Identify the data elements that should appear in a medical record
2. Use a relevant knowledge graph to convert those data elements into specific identifiers
3. Create the phenotype definition by specifying the criteria
4. Iterate by comparison to some reference standard, usually clinician review of the full chart

PROBABILISTIC PHENOTYPING

Problem: Suppose you are the Chief Data Scientist at a healthcare startup. You need to identify patients who have one of fifty different conditions. The work needs to be completed within one week. What could you do?

Solution: Use Supervised Machine Learning with a training dataset, with each patient explicitly labeled as having or not having the condition of interest. The training dataset is used to build a computational model that can classify whether a previously unseen patient has the condition.

Approaches for Probabilistic phenotyping definition:



CONVERT COUNTS OF DIAGNOSIS CODES INTO A PROBABILITY OF A PHENOTYPE

CLUSTERING PATIENTS INTO TWO GROUPS BASED ON DIAGNOSIS CODES.

ASSIGN A PROBABILITY OF HAVING THE PHENOTYPE BASED ON THE NUMBER OF DIAGNOSIS CODES.

Computes the number of times billing codes applied to each patient, and interprets that count as a probability of having the phenotype. Then using that estimated probability, cluster the patients into those likely to have the disease and those not likely.

A mathematical formula:

The expansion of the sample size = 1/(1 - 2t)^2, where t = error rate

Using imperfect data:

1. Start with keywords for the condition of interest
2. Expand the set of keywords to include related terms using a knowledge graph
3. Include all patients who have those keywords as non-negated mentions somewhere on their timeline
4. Then train a classifier using all the other features to classify the phenotype

Another imperfect labeling system uses a concept called "anchors". An **anchor** is a reliable indicator of the presence of the phenotype. The anchor is used as a labeling function to obtain large amounts of training data for a classifier. We can also track how much does our classifier get better with adding more anchors.

Software for Probabilistic Phenotype Definition: Aphrodite (Automated PHenotype Routine for Observational Definition, Identification, Training and Evaluation). It uses standardized, open-source representations: OHDSI CDMv5 and Vocabulary 5, and it implements the labeling. The code is freely available on github: http://github.com/OHDSI/Aphrodite