Storing, Sharing, and Using Algorithms for Implementing Clinical Studies: The Jigsaw Algorithm Repository

Mark D. Danese, MHS, PhD[1], Ryan Duryea[1], Jennifer Duryea, MPH[1], Claire Cangialose[1]
[1]Outcomes Insights, Inc., Westlake Village, CA, USA

Abstract

At the heart of clinical research studies are the small, reproducible units – algorithms – that are combined to retrieve relevant observations from electronic health data. To make research more reproducible and implementation of studies faster, these algorithms should be stored in an unambiguous manner, and readily compiled into a set of reproducible selection statements that operate on the data. The Jigsaw Algorithm Repository is an open-source system designed to accomplish this goal using the OMOP common data model.

Background

Much of computer programming involves breaking large, possibly complicated, processes into small, defined steps that can be accomplished using optimized, tested, and validated code. This facilitates the implementation of reliable and transparent processes. The purpose of the Jigsaw Algorithm Repository (JAR) is to adapt this approach to the implementation of algorithms in research studies using electronic health data.

We define an algorithm as the unique combination of code sets, temporal logic, filters, and database-specific information required to identify records of interest in a clinical database. These algorithms are the smallest repeatable units in any study, and as such, need to contain the minimal amount of information to be unambiguous to researchers, programmers, reviewers, and readers. Algorithms should not include analysis specifications (e.g., definitions of the numerator and denominator for estimating a rate); nor should they include characteristics of individuals that are not specific to the clinical condition identified by an algorithm (e.g., the definition of "breast cancer" should not include gender). These additional details should be specified using separate algorithms that are combined to operationalize a study. For example, an algorithm for "female" plus an algorithm for "breast cancer" might be combined to implement a study of women with breast cancer. This ensures that the same algorithm can be re-used for other studies where different calculations or selection processes may be desired.

Methods

The purpose of the JAR is to allow researchers to store, share and use algorithms. The JAR is an open-source repository that stores algorithms as well as any associated metadata. Metadata might include links to a relevant publication, information about positive predictive values, and details about the populations in which the algorithm was tested. There are 4 classes of algorithms that can be stored currently. They include "tested" algorithms from studies that attempt to validate algorithms against a relevant standard, "published" algorithms that have been defined and used in published studies, and "quality measures" that are developed by a variety of public and private sources.

The JAR also includes "code sets" which define a group of clinically related codes from a specific vocabulary. An example is the Single-Level Clinical Classifications Software (CCS) from the Agency for Healthcare Research and Quality, which divides all ICD-9-CM codes into 285 mutually exclusive groups. Depending on the data, these may or may not be sufficient to be considered "algorithms" per se; this is because with billing (claims) data, code sets are often combined with temporal or other logic to have sufficient specificity. However, because electronic health record systems do not typically record the same diagnosis repeatedly, a simple set of codes may be all that is required to identify relevant records.

The JAR is designed to be used with 3 related applications: the Jigsaw Algorithm Maker (JAM) for constructing or editing algorithms, the Jigsaw Algorithm Viewer (JAV) for visualizing algorithms, and the Jigsaw OBservational Study-builder (JOBS) for assembling algorithms that specify temporal relationships, inclusion criteria, baseline variables, and outcome measures. Within the study builder, algorithms that define each aspect of an observational study are searched and selected. If suitable algorithms are not available, new algorithms can be created with the JAM. When all algorithms are selected, the study builder creates an analysis data set by compiling all of the associated queries. However, because the JAR will be publicly available, it will also possible to adapt it to work with other software.

Underlying the all of these components is the open-source ConceptQL language which is stored in JSON format and can be readily compiled into database-specific SQL statements.[1,2] Combined, these modules allow researchers to design

a study quickly. This approach is facilitated by the use of a common data model (CDM), so that the queries can operate on a known data structure. The current implementation is optimized for OMOP CDM v4. Adaption to OMOP CDM v5 and other data models is under development. The use of the OMOP vocabularies is also important, because it allows researchers to translate codes sets from one vocabulary to another (e.g., ICD-9-CM to SNOMED). This is a feature under development in the JAM to make it easier to adapt algorithms across vocabularies. Finally, the incorporation of algorithms that calculate risk scores is part of the development plan, while algorithms that parse natural language are envisioned as a possible future enhancement.

Current Status

Currently, there are over 2,000 entries included in the JAR including all CCS value sets. These include 26 "tested" ICD-9-CM definitions from United States claims databases based on ICD-9-CM codes, as well as over 100 algorithms for the United Kingdom Clinical Research Practice Datalink (CPRD)[3] that can be used to identify laboratory values, medications, and conditions. Most CPRD condition algorithms are "code sets" derived from published Read codes used to identify Quality Outcomes Framework (QOF)[4] conditions. Between January and September 2015, over 15 analysis data sets have been created using the JAR as a key component of the JOBS. The entire JAR library is currently being re-written in JavaScript to support the JAM and JAV. When finished, the JAR, JAM, and JAV will be made publicly available and the underlying code will be made open-source, with the exception of any non-open source libraries used.

Below is an example of an algorithm to identify congestive heart failure using hospital records from a published validation study. One limitation that should be noted is that the information from the JAR must be inferred from the details of the published study. Therefore, while the algorithm in the JAR is explicitly defined and can be fully implemented, there is no guarantee that it precisely captures the original algorithm in the published study. This is a limitation of methods sections in published studies. The open-access nature of the JAR should allow for the correction and updating of algorithms by the original authors.

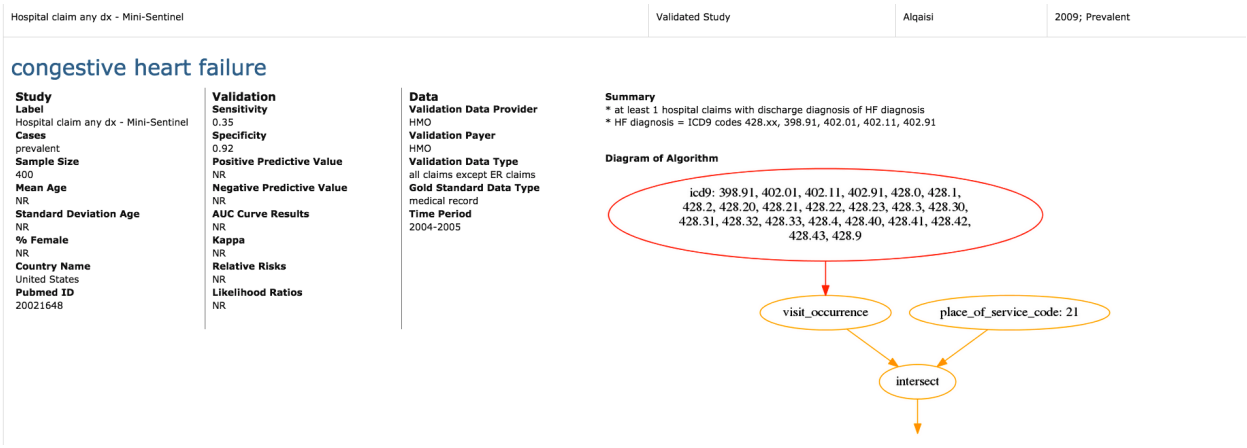

Figure 1: Published algorithm developed to identify congestive heart failure, as listed in the JAR

Conclusion

It is feasible to create, store, and use algorithms to create research studies using electronic health data stored in the OMOP v4 (and soon, v5) CDM. The storage of algorithms in a usable format improves quality, speed and reproducibility of study creation. The OMOP vocabularies enable rapid translation of algorithms across different data sets. The availability of the JAR and its associated components should facilitate development of transparent and reproducible approaches to building studies.

References

1. https://github.com/outcomesinsights/conceptql
2. Duryea R and Danese M. Human Readable Expression of Structured Algorithms for Describing and Storing Clinical Study Criteria and for Generating and Visualizing Queries. AMIA Design Challenge 2015. http://bit.ly/1R7oQY3
3. http://www.cprd.com/intro.asp
4. http://www.hscic.gov.uk/qof