

# A Prototype for Executable and Portable Electronic Clinical Quality Measures Using the KNIME Analytics Platform

Huan Mo, MD<sup>1</sup>, Jennifer A. Pacheco<sup>2</sup>, Luke V. Rasmussen<sup>2</sup>, Peter Speltz,<sup>1</sup> Jyotishman Pathak, PhD,<sup>3</sup> Joshua C. Denny, MD,<sup>1</sup> William K. Thompson, PhD<sup>4</sup>,  
<sup>1</sup>Vanderbilt University, Nashville, TN; <sup>2</sup>Northwestern University, Chicago, IL; <sup>3</sup>Mayo Clinic, Rochester, MN; <sup>4</sup>NorthShore University HealthSystem, Evanston, IL

## Abstract

*Electronic clinical quality measures (eCQMs) based on the Quality Data Model (QDM) cannot currently be executed against non-standardized electronic health record (EHR) data. To address this gap, we prototyped an implementation of a QDM-based eCQM using KNIME, an open-source platform comprising a wide array of computational workflow tools that are collectively capable of executing QDM-based logic, while also giving users the flexibility to customize mappings from site-specific EHR data. To prototype this capability, we implemented eCQM CMS30 (titled: Statin Prescribed at Discharge) using KNIME. The implementation contains value set modules with connections to the National Library of Medicine's Value Set Authority Center, QDM Data Elements that can query a local EHR database, and logical and temporal operators. We successfully executed the KNIME implementation of CMS30 using data from the Vanderbilt University and Northwestern University EHR systems.*

## Introduction

The Quality Data Model (QDM) is an Office of the National Coordinator (ONC)-sponsored standard for representing clinical concepts.<sup>1</sup> It is designed to enable the construction of eCQMs as part of the Meaningful Use (MU) program. The QDM forms the backbone for representing eCQM logical criteria for cohort inclusion and exclusion. The QDM has also been proposed as a standard for constructing executable and portable representations of phenotype algorithms for biomedical research.<sup>2</sup> However, executing QDM-based algorithms currently requires mapping site-specific institutional data to the QDM schema. Although project-specific mappings of institutional clinical data to consensual data dictionaries is frequently feasible (e.g., in the eMERGE Network<sup>3</sup>), to universally and accurately map existing institutional EHR data to the QDM schema and associated terminologies is still a major challenge.

This lack of coupling between QDM-based authoring and local execution platforms is a limiting factor that discourages clinical data analysts from generating QDM-based artifacts. To address this gap, we describe a prototype implementation of the QDM on the Konstanz Information Miner (KNIME, [www.kime.org](http://www.kime.org)). KNIME, an open-source platform with a graphical user interface (GUI), can be used to author executable and portable workflows that are capable of implementing QDM elements and logic. The user-friendly KNIME GUI also allows clinical informaticians with intermediate computational experience to customize QDM-based workflows to use clinical data stored in their local repositories.

## Background

The Centers for Medicare and Medicaid Services (CMS) has sponsored development of the Measure Authoring Tool (MAT) for QDM-based authoring (<https://www.emasuretool.cms.gov>). CMS has also adopted two Extensible Markup Language (XML) formats to represent QDM-based artifacts: Health Quality Measures Format (HQMF), and Simple XML. The QDM is a platform-independent computational model that allows for a variety of implementations. Two open-source implementations that are built on the QDM include popHealth, which executes the eCQM logic against imported patient data, and Cypress, which is used to validate the execution of a quality measure system.

In the QDM, clinical events are represented as codes from an approved set of standard terminology systems, such as the International Classification of Diseases (ICD) or Logical Observation Identifiers Names and Codes (LOINC). For each data element, usually more than one code is considered appropriate. For example, the element of *an active diagnosis of acute myocardial infarction* (AMI) accepts twenty codes in ICD Ninth Revision, Clinical Modification (ICD-9-CM). These collections of codes are called *value sets*. The QDM uses unique object identifiers (OIDs) to refer to value sets that are stored and published by the National Library of Medicine's Value Set Authority Center (VSAC,

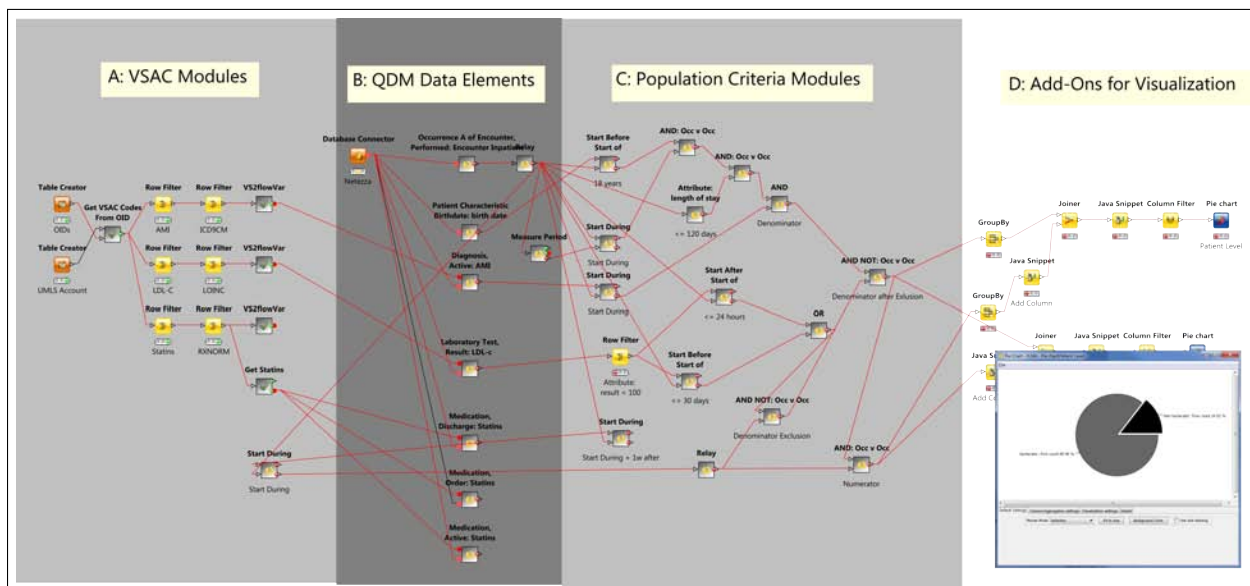


Figure 1: Implementation of CMS30 (Statin at Discharge) on KNIME. Arrow-headed lines denote transmission of data tables; square-headed lines denote database connections; round-headed lines denote transmission of flow variables. Pie chart in lower right is generated from visualization nodes in Region D.

<https://vsac.nlm.nih.gov>). VSAC provides a web service application programming interface (API) for local applications to retrieve value sets by using their assigned OID.

KNIME Analytics is an open source platform that integrates data access, data transformation, statistical analysis, data-mining tools, and snippets of different programming languages in a visual workbench. KNIME computing is based on workflows for tables and flow variables. Transformation steps are performed by sets of *nodes*, which are connected together into a workflow *graph*. Each node accepts one or more tables and flow variables as input, executes a unit of computation, and generates flow variables and tables as output. Users can collapse a series of nodes into a single *metanode* to represent a functional group. Developers can also extend KNIME with new sets of nodes using the plug-in API. The parameters of nodes are easily configured with pop-up windows, and after executing a node users can check the associated output table(s). KNIME also supports looping and conditionals for branching workflows.

To evaluate the suitability of KNIME to handle QDM-based artifacts, we chose to implement eCQM CMS30 (Version 2, NQF #0639, titled: Statin Prescribed at Discharge,<sup>4</sup> developed by the Oklahoma Foundation for Medical Quality). We selected this eCQM because it uses widely available EHR elements (e.g., inpatient encounters, diagnosis codes, medication, laboratory results) as well as various logical and temporal operators. It measures the proportion of patients that are prescribed with statin drugs at discharge among admitted patients for AMI. To test portability, we executed the CMS30 KNIME workflow on both Vanderbilt University (VU) and Northwestern University (NU) data. We then performed an informal analysis of the results.

## Methods

Figure 1 shows the overall KNIME workflow for CMS30. It can be conceptually divided into four regions, as shown in the figure. Here we focus on the elements most relevant to implementing the QDM.

### Retrieving and Transforming Value Sets

Region A of Figure 1 contains KNIME metanodes to retrieve value sets from NLM's VSAC through its web service API, using a list of value set OIDs (derived from CMS30) and the user's Unified Medical Language System (UMLS) account information (required by VSAC for ticketing).

For the convenience of downstream usage, we transform the codes in a value set to a string ready to plug into a Structured Query Language (SQL) clause for queries to institutional clinical data repositories. In this step, we encountered

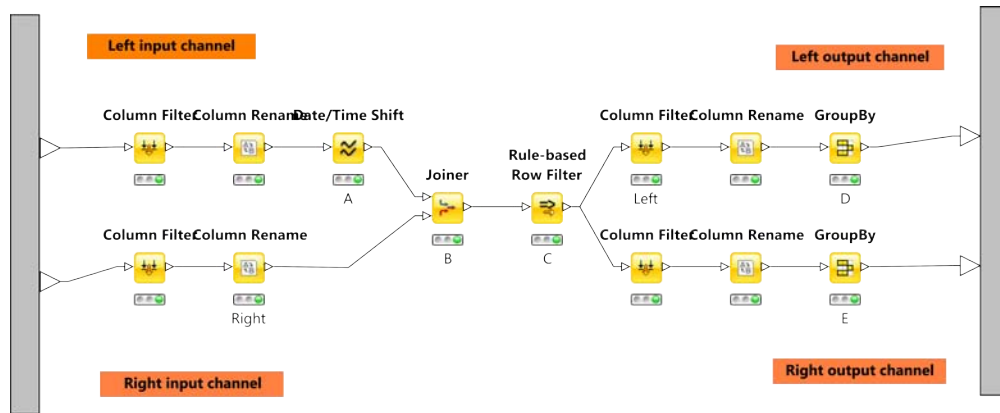


Figure 2: Implementing the temporal operator ( $\leq 30$  day(s) starts before start of)

three situations. In the simplest case, diagnosis codes in VU and NU are fully compatible with the ICD-9-CM value set extracted from CMS30, so we transform the value set to a string to plug into a SQL IN clause. In a slightly more complicated case, we must first create a KNIME mapping from local VU and NU laboratory codes to LOINC codes, which KNIME then joins to the LOINC value set extracted from CMS30. In the most challenging case, while CMS30 uses RxNORM concepts for fully complete clinical drug names (ingredient, strength, and dose form), medication lists in both VU and NU have not been standardized to match these concepts directly. To solve this discrepancy, we acquired the ingredient and brand names from the CMS30’s medication concepts with the RxNORM web service (<http://rxnav.nlm.nih.gov/RxNormRestAPI.html>), and formed them into a string for a regular expression clause. In all three cases, we pass resulting strings of value sets to KNIME flow variables.

### Generating QDM Data Elements

Region B of Figure 1 contains KNIME metanodes that implement QDM Data Elements. A QDM Data Element includes a data type (e.g., `DiagnosisActive`), a value set, and a set of attributes (e.g., `start/end datetime`, `result`). The data types imply EHR fields (or tables in an IDR), and attributes imply the columns of the IDR tables. We implement each QDM Data Element with a database-querying module (using a the value set strings from flow variables described above) and an adapting module to standardize columns into QDM format. These elements form the primary locus of customization to site-specific EHR data across institutions.

The central Data Element in CMS30 is “Occurrence A of Encounter Performed: Encounter Inpatient” (`OccA-EP`). Every other Data Element is first temporally related to the `OccA-EP` set (table), and all downstream logical operators (`AND`, `AND NOT`, `OR`) are applied to subsetted `OccA-EP` events (rows) resulting from the temporal operations. Every `OccA-EP` event is associated with the attributes `pid` (patient identifier), `StartDateTime`, and `EndDateTime`.

### Applying QDM Functions and Operators

Region C of Figure 1 contains implementations of QDM logical elements that operate over (attributes) of QDM Data Elements. After data for QDM Data Elements are imported, QDM Population Criteria employ logical and temporal operations with these data to define different populations (Initial Patient Population, Denominator, Denominator Exclusions, Numerator, Denominator Exceptions). These operations are performed in KNIME by creating metanodes implementing the corresponding QDM logic. We describe the implementation of some important operators.

*Temporal Operator* ( $\leq 30$  day(s) starts before start of): This temporal operator (see Figure 2) takes two input tables (denoted with *left* and *right*) each with at least two fields (`pid` and `StartDateTime`). First, we create a new column (called `LeftAdded`) on the left input table with 30 days added to their `StartDateTime` (Node A on Figure 2). Then we inner join the left and right input tables with `pid` (Node B), and filter the rows where the `StartDateTime` (left) is earlier than `StartDateTime` (right) AND `LeftAdded` is not earlier than `StartDateTime` (right) (Node C). Then, we split the joined table back to the left and right output tables (corresponding to input tables) with removing duplicated rows (Node D and E).

*Logical operators* (AND, AND NOT, OR): These are implemented with the operations shown in Table 1. For each instance of these operators, two customizations need to be addressed. First, logical operators mostly use outputs from temporal operators, which have two output tables corresponding to their two input tables, so the logical operators need to select the semantically correct table to use. Second, algorithms for logical operators require carefully selecting appropriate input columns in the joining steps, depending on whether they are operating on event-level data or patient-level data.

Operator	Algorithm Steps
AND	<ol style="list-style-type: none"> <li>1. Inner join two input tables using appropriate columns;</li> <li>2. Filter columns for the output table;</li> <li>3. Remove duplicated rows (with <code>GroupBy</code>).</li> </ol>
AND NOT	<ol style="list-style-type: none"> <li>1. Add a column with non-null values to right-hand-side input table;</li> <li>2. Left outer join two input tables using appropriate columns;</li> <li>3. Filter and keep rows that have null in the added column from step 1;</li> <li>4. Filter columns for the output table and remove duplicated rows.</li> </ol>
OR	<ol style="list-style-type: none"> <li>1. Filter appropriate columns from both input tables;</li> <li>2. Concatenate the two trimmed input tables and remove duplicated rows.</li> </ol>

Table 1: Algorithms used in implementing QDM logical operators

### Test Data

At VU, we tested this implementation with the Synthetic Derivative (SD) database,<sup>5</sup> which contains de-identified clinical information derived from VU’s EHR systems. The SD database contains clinical and demographic information, including as diagnosis codes, procedure codes, medications, laboratory values, and de-identified clinical documents and reports. It can be used as a stand-alone research resource, or in conjunction with BioVU to identify record sets for genome-phenome analysis. The execution was carried out on a desktop environment, so we used the subset of VU’s genotyped population of 35,842 patients, with the measure period set at 2005-05-01 and ending at 2012-04-30.

At NU, we tested this implementation with data stored in the Northwestern Medicine Enterprise Data Warehouse (NMEDW) (<http://informatics.northwestern.edu/blog/edw>). The execution was also carried out on a desktop environment, so we used the subset of NU’s genotyped population of 4,838 patients, with the same temporal period used for VU. The NMEDW contains clinical and demographic data aggregated primarily from two commercial EHR systems (Cerner and Epic). It also contains basic information on research subjects, including those enrolled in NU’s NUGene DNA biobank.

### Results

The KNIME-based workflow was successfully executed against VU’s SD data as planned, by simply configuring the data access and mapping nodes (Figure 1, Regions A and B). Execution of eCQM CMS30 resulted in a final score of 35% of VU patients satisfying the measure requirement (see below for further analysis of this score). Porting the implementation of eCQM CMS30 to NU’s EDW was straightforward, involving also the solitary step of configuring the data element and mapping nodes. Execution at NU resulted in a final score of 85% of NU patients meeting the measure. Although this is just a single eCQM prototype, this provides evidence that KNIME workflows are a promising approach to implementing QDM-based logic, portable with minimal effort across multiple institutions.

One of our primary objectives was to develop a solution that not just implements QDM-based logic, but also gives us the flexibility to examine and modify the results, on a per institution basis. KNIME proved capable of satisfying this goal, by providing a very flexible platform with a full array of computational tools for extending measures with downstream analysis (e.g., machine learning, statistics, graphical reports) that is not part of the QDM per se. It is also quite

easy to adjust the connections between KNIME nodes and/or add additional nodes to provide further customization for end users. For example, CMS30 uses statins as discharge medication, and our final score for VU is only 35%. However, if we adjust the KNIME workflow to use statins as ordered medication during the inpatient encounter, the VU score increases to 80%. These two situations are virtually equivalent, and the discrepancy in scores arise mostly from variation in the documentation that we searched as part of this exercise – excluding one of the most common discharge notes in the first analysis. (A score of 80% is still lower than expected, perhaps due to specific exclusions, inclusion errors, or hospital transfers.) At NU, 85% of the patients met the measure, when using medications ordered during inpatient stays, the closest surrogate to medications ordered at discharge. The score remained the same even when expanded to all ordered medications, including those ordered during outpatient encounters.

## Conclusion

In developing this prototype, we address the current gap between the specification of QDM-based eCQMs and their application to existing clinical data. Continuing work on standardizing EHR systems is imperative to ensuring the efficacy of secondary use of future EHR data. At the same time, the currently available EHR data are an invaluable resource for the design and validation of QDM-based measures and algorithms.

There are a few limitations to the work described here. First, the utility of our approach would be greatly increased if there was a tool to automatically translate HQMF eCQMs into KNIME workflows. We are currently in the process of implementing such a translation tool, and will describe this in future work. This tool will immediately and greatly expand the number of eCQMs that can be executed as portable KNIME workflows. Second, our current prototype implementation uses KNIME metanodes, which may be more difficult for users to adapt to new measures and local environments. We plan to develop a complete QDM extension package for KNIME to provide native QDM nodes for a better user experience. Third, data for some QDM Data Elements (e.g., medication order not done) and attributes (e.g., discharge status) are not readily available in either VU or NU integrated data marts, and their implementation in KNIME requires future work. Fourth, we do not eliminate the need for informaticians with basic skills in SQL and programming to author and adapt measures and algorithms. Instead, our goal for this project is to enable these informaticians to use the QDM with their own projects, and to share executable artifacts with external institutions. Enabling of such a user community is important for the evolution and adoption of the QDM.

## Acknowledgements

This work has been supported in part by funding from PhEMA (R01 GM105688) and eMERGE (U01 HG006379, U01 HG006378 and U01 HG006388).

## References

- [1] Quality Data Model, Version 4.1.1. Centers for Medicare & Medicaid Services Office of the National Coordinator for Health Information Technology; 2014. Available from: [http://www.healthit.gov/sites/default/files/qdm\\_4\\_1\\_1.pdf](http://www.healthit.gov/sites/default/files/qdm_4_1_1.pdf).
- [2] Thompson WK, Rasmussen LV, Pacheco JA, Peissig PL, Denny JC, Kho AN, et al. An evaluation of the NQF Quality Data Model for representing Electronic Health Record driven phenotyping algorithms. In: AMIA Annual Symposium Proceedings. vol. 2012; 2012. p. 911.
- [3] Kho AN, Pacheco JA, Peissig PL, Rasmussen L, Newton KM, Weston N, et al. Electronic Medical Records for Genetic Research: Results of the eMERGE Consortium. *Science Translational Medicine*. 2011 Apr;3(79):79re1–79re1.
- [4] Krumholz HM, Anderson JL, Bachelder BL, Fesmire FM, Fihn SD, Foody JM, et al. ACC/AHA 2008 performance measures for adults with ST-elevation and non-ST-elevation myocardial infarction: a report of the American College of Cardiology/American Heart Association Task Force on Performance Measures. *Journal of the American College of Cardiology*. 2008 Dec;52(24):2046–2099.
- [5] Roden DM, Pulley JM, Basford MA, Bernard GR, Clayton EW, Balser JR, et al. Development of a large-scale de-identified DNA biobank to enable personalized medicine. *Clinical Pharmacology and Therapeutics*. 2008 Sep;84(3):362–369.