# ScriptNumerate: A Data-to-Advice Pipeline using Compound Digital Objects to Increase the Interoperability of Computable Biomedical Knowledge

**Allen J. Flynn, PharmD, PhD[1,3], Julia Adler-Milstein, PhD[1,4], Peter Boisvert, BA, MFA[3], Nate Gittlen, BS[3], Carl Lagoze, PhD[1], George Meng, MSEE[3], F. Jacob Seagull, PhD[3], Charles P. Friedman, PhD[1,2,3]**
**[1]School of Information, University of Michigan, Ann Arbor, MI;**
**[2]School of Public Health, University of Michigan, Ann Arbor, MI;**
**[3]Medical School, University of Michigan, Ann Arbor, MI;**
**[4]School of Medicine, University of California San Francisco, San Francisco, CA**

**Abstract**

*Many obstacles must be overcome to generate new biomedical knowledge from real-world data and then directly apply the newly generated knowledge for decision support. Attempts to bridge the processes of data analysis and technical implementation of analytic results reveal a number of gaps. As one example, the knowledge format used to communicate results from data analysis often differs from the knowledge format required by systems to compute advice. We asked whether a shared format could be used by both processes. To address this question, we developed a data-to-advice pipeline called ScriptNumerate. ScriptNumerate analyzes historical e-prescription data and communicates its results in a compound digital object format. ScriptNumerate then uses these same compound digital objects to compute its advice about whether new e-prescriptions have common, rare, or unprecedented instructions. ScriptNumerate demonstrates that data-to-advice pipelines are feasible. In the future, data-to-advice pipelines similar to ScriptNumerate may help support Learning Health Systems.*

**Introduction**

To maintain and extend their clinical decision support (CDS) capabilities, organizations must update and upgrade the machine-interpretable (i.e., computable) biomedical knowledge used by Electronic Health Records (EHRs) and other CDS systems. This knowledge needs to be kept current to ensure that CDS gives advice that is well-informed. To extend their scope, CDS improvements reported in the literature need to be confirmed, and, when that is done, scaled up to benefit more people[1,2]. Yet sometimes promising new CDS capabilities reported in the literature are lost when systems are replaced[3]. To maintain and extend CDS more effectively, better ways of managing the *computable biomedical knowledge* (CBK) undergirding CDS are needed[4]. Further, as health organizations begin to gain new knowledge by aggregating and analyzing real-world data, causing the volume, velocity, and variety of new knowledge production to increase, the need for more effective CBK management is likely to grow as well.

CDS, Learning Health Systems, and indeed all advice-giving systems for health are dependent on CBK. Advice-giving systems draw on previously collected CBK, apply it to data about the context of a current decision problem to generate advice, and then communicate the advice they generate to decision makers.

CBK is represented in machine-interpretable forms to make it useable by advice-giving systems. There are many ways to represent it so that it is executable by machines. Today, advice-giving systems often use CBK that is represented and formatted in unique ways and stored in proprietary knowledgebases[5]. Consequently, this CBK is not interoperable with a wide array of different advice-giving systems. In part because of its unique format, it is more difficult to maintain, extend, and aggregate proprietary forms of CBK[5,6]. Overall, the additional work required to do these things with CBK in proprietary formats limits the frequency and scope of advice-giving system upgrades and the advances that could come by extending the scope and scale of the CBK that these systems use[5].

To make CBK more widely accessible to various advice-giving systems and easier to maintain, extend, and aggregate, we see opportunities to increase its interoperability at its inception. One of these opportunities is to use *compound digital objects* to package CBK[7]. Compound digital objects are formally structured digital objects. They are modular and, by definition, they each have a unique identifier[7]. Here we report experiments to develop and trial a data-to-advice pipeline called ScriptNumerate that features compound digital objects. ScriptNumerate uses a common object format both to communicate the analytic results it generates and also to apply those results, which constitute CBK, to produce advice. For these experiments, ScriptNumerate builds on a new knowledge management and deployment platform for CBK that we are developing. We call this new platform the Knowledge Grid (www.kgrid.org).

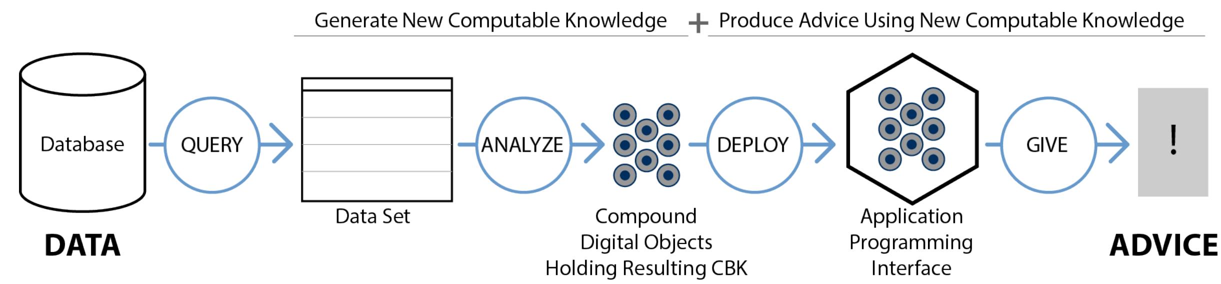


**Figure 1.** General case of a Data-to-Advice Pipeline that features compound digital objects for better CBK interoperability

Figure 1 above portrays the general case of a data-to-advice pipeline. The pipeline combines four processes to generate new computable knowledge and then use it to produce advice. Starting on the left with a Database, the first process runs a query to get a Data Set. The second process analyzes that Data Set using a method that produces compound digital objects holding the results as machine-interpretable CBK. This analytic process could involve many different data analysis methods of varying complexity. The third process then deploys those same objects via an Application Programming Interface (API) mechanism. The fourth and final process uses the API to give advice.

**Background and Significance**

We begin by noting some prior research on interoperability. Next, we differentiate this study from related efforts to increase health information resource interoperability. We end by discussing electronic prescriptions or *e-prescriptions*, the RxNorm terminology, and our past work to systematically alert prescribers to atypical e-prescriptions.

*Interoperability*

Here we define interoperability to be the *ability for arbitrary things to function jointly*[8]. This definition clarifies that interoperability is not something things either have or do not have (although it is sometimes described that way). Instead, interoperability is an ability that varies by degrees in the things that exhibit it.

Our primary interest is increasing the interoperability of CBK so that advice-giving systems can be upgraded with new or revised CBK more easily. To this end, we believe gains are possible by leveraging how the architecture and methods of the World Wide Web (WWW) enable APIs that enhance information resource interoperability in general[9].

*Increasing interoperability*

When the WWW was formative, Paepcke et al.[10] described five clusters of approaches comprising the solution space for increasing the interoperability of information resources. These clusters are the column headers of Table 1 below.

**Table 1.** Five clusters of approaches for increasing interoperability of information resources[10]

| | **Strong standards** | **Families of standards** | **External mediation** | **Specification-based API interactions** | **Mobile functionality** |
|---|---|---|---|---|---|
| Examples | HTML, FHIR, HTTP, JSON, TCP/IP, OAuth | Family of Scripting Computer Languages | PDF to Text file converter | APIs described using the OpenAPI Specification | JavaScript modules |

Paepcke et al. argue that, to improve interoperability, multiple approaches need to be pursued in light of known tradeoffs[10]. They recount these tradeoffs: Strong standards are helpful but hard to develop, institute, and maintain. Families of standards offer flexibility at the price of greater variability and higher support costs. External mediation between competing standards usually results in some information loss. Specification-based API interactions are useful, but also complex. Finally, mobility is facilitated by modularization and containerization of information resources. We use approaches from all five clusters to increase the interoperability of the CBK used in ScriptNumerate.

More recently, Van de Sompel and Nelson[9] reported on 15 years of interoperability efforts for WWW-based systems. They point out a key lesson learned, which is that the technical methods of the WWW's "uniform interface" for webservice APIs (i.e., GET, POST, PUT, DELETE), combined with typed hyperlinks, defined media types, and the Resource Description Framework (RDF), can significantly increase the interoperability of compound digital objects[9].

*Increasing interoperability of health information resources*

Many initiatives already underway to enhance interoperability for health information resources relate to our work. Health Level 7's Fast Health Interoperability Resources (HL7 FHIR) is a prime example. Efforts to create FHIR are described elsewhere[11]. FHIR directly supports health information exchange as a strong data resource standard that builds on other general technical standards (e.g., JSON) and can incorporate established health IT terminology standards (e.g., LOINC, RxNorm, SNOMED).

The *CDS Hooks* project is another notable interoperability effort[12]. In CDS Hooks, each "hook" specifies a particular technical interaction between an EHR and an external resource, whereby the EHR requests and receives advice for clinical decision support from the external source. In this manner, CDS Hooks enables all EHRs to share contextual information in a similar way at moments when predefined events trigger EHRs to call external systems for advice.

The Substitutable Medical Applications and Reusable Technologies (SMART) project is advancing towards an open ecosystem of apps that extend and specialize the capabilities of EHRs[13]. SMART brings together FHIR, OAuth2 (an authorization standard), OpenID Connect (an authentication standard), Hypertext Markup Language (HTML), and CDS Hooks to create shareable and interoperable "SMART apps" that interoperate with multiple EHR platforms.

Our own ScriptNumerate uses the Knowledge Grid (KGrid), a platform under development in our lab for managing and deploying CBK. In KGrid, an instance of CBK is an executable representation of biomedical knowledge, such as a computable guideline. KGrid formally specifies a class of compound digital objects called Knowledge Objects[14]. A Knowledge Object is a specially formatted digital "package" combining an instance of CBK with descriptive metadata and other relevant information[15]. KGrid also provides a technological component, called the ***Activator***. The KGrid Activator makes CBK held in Knowledge Objects available for use via an WWW-based API.

Work on KGrid is complementary to FHIR, CDS Hooks, and SMART apps. KGrid's Knowledge Objects support an API that already accepts structured data resources, like FHIR resources, as inputs and also generates them as outputs. In addition, CDS Hooks and SMART apps can draw on CBK through KGrid's API.

*Structured form of electronic prescriptions*

As a byproduct of the widespread adoption of EHRs, most prescriptions in the U.S. are now e-prescriptions[16]. E-prescriptions are digital text objects represented using either proprietary or open data schemas, including HL7's FHIR *MedicationRequest* resource schema and the National Council for Prescription Drug Program's *SCRIPT* schema. These schemas define the discrete data elements that comprise parts of an e-prescription. These parts include a named drug product, its active ingredients and their quantities, and the e-prescription *Sig*, a Latin term for a *label,* which indicates the *instructions for use* of a drug product (e.g., *Take 100mg every morning*).

ScriptNumerate is so named because it automatically counts historical e-prescription *Sigs* to generate *statistical Sig frequencies*. It uses these statistical frequencies to compute advice telling if new e-prescriptions have atypical Sigs.

The e-prescriptions used as a data source for this project are represented using a proprietary data schema of one EHR vendor. This proprietary schema represents e-prescriptions using the following seven discrete data elements:

- local unique drug product identifier (e.g., "310")
- drug product name (e.g., acyclovir 400mg tablet)
- generic ingredient name (e.g., acyclovir)
- ingredient strength (e.g., 400 mg)
- dosage form (e.g., tablet)
- route of administration (e.g., oral)
- Sig (e.g., take 400 mg 2 times daily)

*Use of RxNorm*

RxNorm is a terminology from the National Library of Medicine[17]. It enables e-prescription data to be normalized. RxNorm assigns Concept Unique Identifiers called RXCUIs to drugs and drug products. The *Semantic Clinical Drug* (SCD) subclass of these identifiers pertains to this project. Every SCD in RxNorm has three terms, **active ingredient** + **strength** + **dosage form**. For example, '*Acyclovir + 400mg + Oral Tablet*' is an RxNorm SCD with RXCUI = 197311. RxNorm SCDs are used by ScriptNumerate to normalize locally acquired e-prescrption data at the *drug product* level.

*Prescription atypicality advice*

As part of their routine practice, pharmacists review prescriptions prior to medication dispensing[18]. Not long ago, when most prescriptions were handwritten, pharmacists had to carefully interpret physician handwriting. Today's e- prescriptions are completely legible[16]. As a result, the task of prescription review is changing.

Presently, when reviewing e-prescriptions, pharmacists must determine whether each e-prescription is safe, likely to bring therapeutic benefits, and cost effective. To help them do this, pharmacy information systems provide alerts for allergies, drug interactions, and cost[19]. However, medications with highly similar names, the increasing annual volume of e-prescriptions, and the large number of alerts make it difficult to spot every problem. As a result, odd, atypical, and unwarranted e-prescriptions are sometimes inadvertently approved by pharmacists. This leads to harm[20].

To help prevent harm, information technology can assist pharmacists and others to detect odd or *atypical* e-prescriptions[21]. Although this is not yet commonly being done, the ready availability of historical e-prescription data enables comparison of newly created e-prescriptions with past e-prescriptions. In 2014, we published a method of using historical e-prescription information to detect and highlight rare or unprecedented, i.e., *atypical e-prescriptions*[21]. On a limited basis, we trialed a new alert for atypical e-prescriptions. Unfortunately, the capability to display atypical e-prescription alerts was later lost at our site due to an EHR replacement.

Comments on atypical e-prescription alerts have since appeared in the scientific literature. Kane-Gill highlights the low frequency of e-prescriptions triggering these alerts (0.4%) during our initial study[22]. Yet Ferrández et al. correctly note that, in that same study, atypical e-prescription alerts were configured to fire for only five high-risk medications[23].

The limited scope of our initial study, and the loss of our capability to generate atypical e-prescription alerts, motivated us to develop ScriptNumerate by using compound digital objects holding CBK and technical components of KGrid.

*Significance*

ScriptNumerate is significant for improving medication safety. It builds on the previously established potential for recognizing atypical e-prescriptions systematically[21]. It offers a new means to expand both the scope and the scale of this capability. We recognize, however, that screening for atypical e-prescriptions is just one of several core capabilities needed to identify problematic prescriptions and ultimately partially automate e-prescription review[18].

Yet the greater significance of this work may be that it demonstrates how a data analysis pipeline can generally be extended to become a data-to-advice pipeline (Figure 1). We believe this is significant because data-to-advice pipelines have the potential to better support health organizations by increasing the interoperability of CBK[24].

## Research Questions

The following three research questions were investigated.

***RQ1.*** What are the required technical functions of a data-to-advice pipeline capable of computing advice about atypical e-prescription Sigs based on historical e-prescribing patterns?

***RQ2.*** What types of atypical e-prescription Sig advice, and how much of each type, can be generated by using ScriptNumerate to compare contemporary e-prescriptions for an elderly adult population of hospital inpatients with historical e-prescribing patterns arising from treatment of a similar population during the previous year?

***RQ3.*** How can ScriptNumerate aggregate CBK about historical e-prescriptions from multiple EHR sources to form combined collections of Sig Frequency Knowledge Objects?

## Methods

*Conceptual overview of ScriptNumerate*

The ScriptNumerate pipeline is portrayed conceptually in Figure 2 below. Starting on the left, as a byproduct of prescribing, e-prescription data is collected in an *EHR database*. Those data are queried to form a *table of historical e-prescriptions*. That tabular data set is analyzed, on a drug product specific basis, in a manner analogous to making a pivot table, resulting in counts of unique Sigs by drug product. These Sig counts are represented in the Python computer language to constitute a simple form of CBK about prescribing patterns. As shown in Figure 2, this CBK is stored in numerous *drug product specific Sig frequency files*, which are *passive knowledge resources*.

Importantly, the pipeline continues beyond this analytic endpoint by converting passive *drug product specific Sig frequency files* into *active advice-giving services*. This happens via ScriptNumerate's API, which enables on-demand interrogation of the CBK in its *drug product specific Sig frequency files*. As the two green arrows on the right side of Figure 2 indicate, advice-giving services provided by this API are engaged to generate advice indicating whether new e-prescription Sigs are common, rare, or unprecedented compared to past prescribing practices.

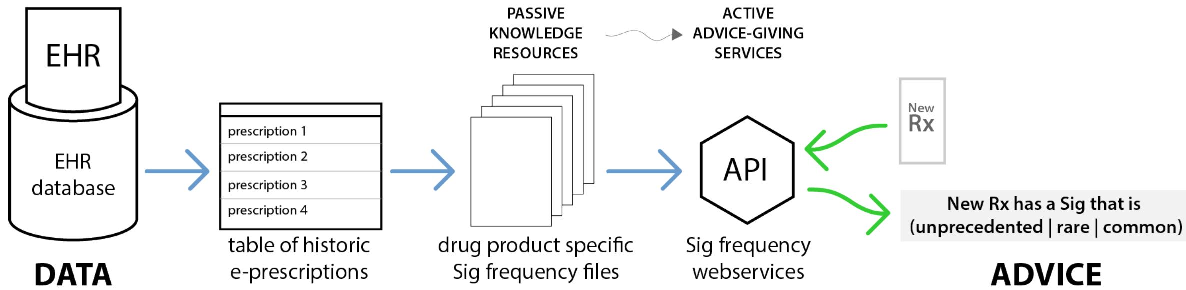


**Figure 2.** Concept of ScriptNumerate: A Data-to-Advice pipeline for E-prescription Sig Frequency Advice

*Use of KGrid*

This study uses the KGrid platform to help convert passive CBK resources into active CBK-enabled advice-giving services. Using KGrid's technology, a special class of *compound digital objects* holding CBK, called Knowledge Objects (KOs), are automatically created and then deployed within ScriptNumerate to engender *advice-giving services*. These KOs are built in accordance with a formal specification that we developed[14]. Once they are built automatically, ScriptNumerate's KOs are then "activated" by loading them into a deployed instance of the KGrid Activator. The Activator provides the API that gives rise to ScriptNumerate's advice-giving capabilities. The Activator oversees computer execution of the CBK held in the KOs. By using KGrid's Activator component, ScriptNumerate's API can process e-prescription data and compute e-prescription-specific advice.

*Software development for this study*

To build the additional software pieces of ScriptNumerate needed to complement available components of the KGrid, a set of software scripts were written using PERL (5.18.2) and Python (2.7.10). In addition, we used Python and JavaScript (ECMAScript 5) to encode CBK for KOs later made serviceable by KGrid's Activator.

*Data source and data characterization*

For this study, after gaining approval from the Institutional Review Board of the University of Michigan, we used two *limited data sets* each containing one year's worth of e-prescriptions. Every e-prescription had a drug product associated with an RxNorm SCD code. Every e-prescription was actually placed in the EHR for a hospital inpatient age 75 years or older. We ran queries to extract these historical e-prescription data from the University of Michigan's EHR data repository (Clarity, EpicCare, Epic, Verona, WI.).

A sub-population of elderly adults, age 75 and older, was chosen for these experiments because drug dosages, and hence e-prescription Sigs, are specific to this subpopulation[25]. In practice, comparisons of e-prescription Sigs to historical prescribing patterns stratified by subpopulation are required because prescribing is population-specific.

The first data set included e-prescriptions from calendar year 2016 (n = 232,203). From these data, historical *prescribing patterns*, in the form of statistical Sig frequencies, were determined on a drug product basis for a population of elderly adults. A historical e-prescription prescribing pattern was generated for the 431 drug products that were prescribed at least weekly, on average, for this population during 2016. E-prescriptions for drug products prescribed less often are treated as *e-prescriptions for infrequently prescribed drug products* by ScriptNumerate.

Figure 3 below shows two graphs of counts, by drug product, of the number of e-prescriptions in the 2016 data set. A total of 1819 unique drug products are included. The y-axes have a *log scale*. The graph on the left shows the number of e-prescriptions placed for 1388 drug products prescribed fewer than 50 times for the study population in 2016. Note that 332 drug products were only prescribed *once* in 2016 for this population. The graph on the right shows the number of e-prescriptions registered for 431 commonly prescribed drug products. Of these, 129 drug products were prescribed between 50 and 100 times in 2016 for the study population, while 302 others were prescribed more than 100 times.

The second e-prescription data set, spanning all of calendar year 2017, was similar to the first. It included e-prescriptions for the same 1819 unique drug products (n = 251,928). Using ScriptNumerate's API, every e-prescription from 2017 was automatically compared to the historical prescribing patterns derived from the previous year's e-prescription data. The 2017 data were processed automatically to ascertain, for all e-prescriptions for 431 commonly prescribed drug products (n = 236,320), whether or not each e-prescription's Sig was common, rare, or unprecedented in comparison to the Sigs for the same drug products from 2016. ScriptNumerate also advised in all cases when an infrequently prescribed drug product was prescribed in 2017 (n = 15,608).

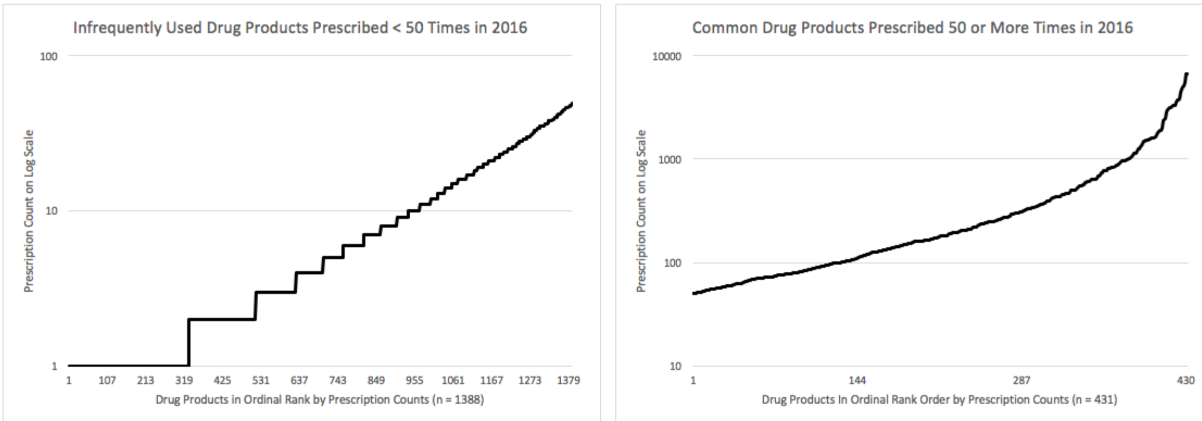


**Figure 3.** Prescription counts per rank-ordered drug product from 2016 for hospital inpatients 75 years and older are given on a log scale for 1388 infrequently prescribed and used drug products (left) and 431 commonly prescribed drug products (right)

*Generation and quantification of advice*

We used a software script to compute advice for 251,928 e-prescriptions from 2017 using ScriptNumerate. This batch process took 80 minutes to run on a laptop computer. We used a spreadsheet to quantify the number of e-prescriptions with unprecedented, rare, or common Sigs. Because the definition of a rare Sig is arbitrary, we chose two different thresholds to define what a rare Sig is. Common Sigs are also arbitrary and are defined as the obverse of rare Sigs.

## Results

To begin this section, results are given for a single drug-product to explain and clarify what ScriptNumerate does. The exemplar drug product used is *Acyclovir 400mg Oral Tablet*, an antiviral drug product.

**Table 2.** Results from using ScriptNumerate only for the drug product *Acyclovir 400mg Tablet* (Rare is <=10% of previous Sigs)

| Count of 2016 E-prescriptions | Six Sigs used for Acyclovir 400mg Oral Tablet in 2016 (with corresponding statistical frequencies) | | Results from comparing 190 2017 e-prescription Sigs | | |
|---|---|---|---|---|---|
| | | | Unprecedented | Rare | Common |
| 141 | 400 MG 2 TIMES DAILY (123)<br>400 MG 3 TIMES DAILY (4)<br>400 MG EVERY 12 HOURS (2) | 400 MG ONCE DAILY (8)<br>400 MG 5 TIMES DAILY (3)<br>800 MG ONCE DAILY (1) | 10 | 21 | 159 |

As shown above in Table 2, 141 historical e-prescriptions for this product were placed during 2016 with six different Sigs. The most common Sig was "400 MG 2 TIMES DAILY", which was used for 123 of 141 e-prescriptions. ScriptNumerate stores these six Sigs and their frequencies in a drug product Sig frequency file. This file is then encoded in Python, making it an instance of CBK. Next, using ScriptNumerate's API and this instance of CBK, 190 e-prescriptions from 2017 were processed to assess their Sigs. The results show that, of these 190 e-prescriptions, 159 had a common Sig, 21 others had a rare Sig, and 10 more had unprecedented Sigs, which are Sigs that do not appear at all in the list of Sigs from 2016. An example of an unprecedented Sig from 2017 is "200 MG 5 TIMES DAILY." In this case, instead of prescribing half of a 400 mg oral tablet, there is a 200 mg tablet that could be prescribed.

***RQ1.*** *What are the required technical functions of a data-to-advice pipeline capable of computing advice about atypical e-prescription Sigs based on historical e-prescribing patterns?*

Figure 4, found below, is a technical depiction of the ScriptNumerate data-to-advice pipeline. It portrays nine essential components required for the ScriptNumerate data-to-advice pipeline to function as intended. These are described next.

The first technical function required in the ScriptNumerate pipeline, illustrated in Figure 4 and indicated by ①, is to run a **SQL query** against an EHR database resulting in a table of historical e-prescriptions from the EHR.

The next required function, indicated by ②, is the **rxcounter**. It does what a "pivot table" function does by automatically generating counts, specific to each drug-product, resulting in the frequencies of the Sigs used in e-prescriptions for that drug product. The result of the **rxcounter** function is a series of drug product specific Sig frequency text files.

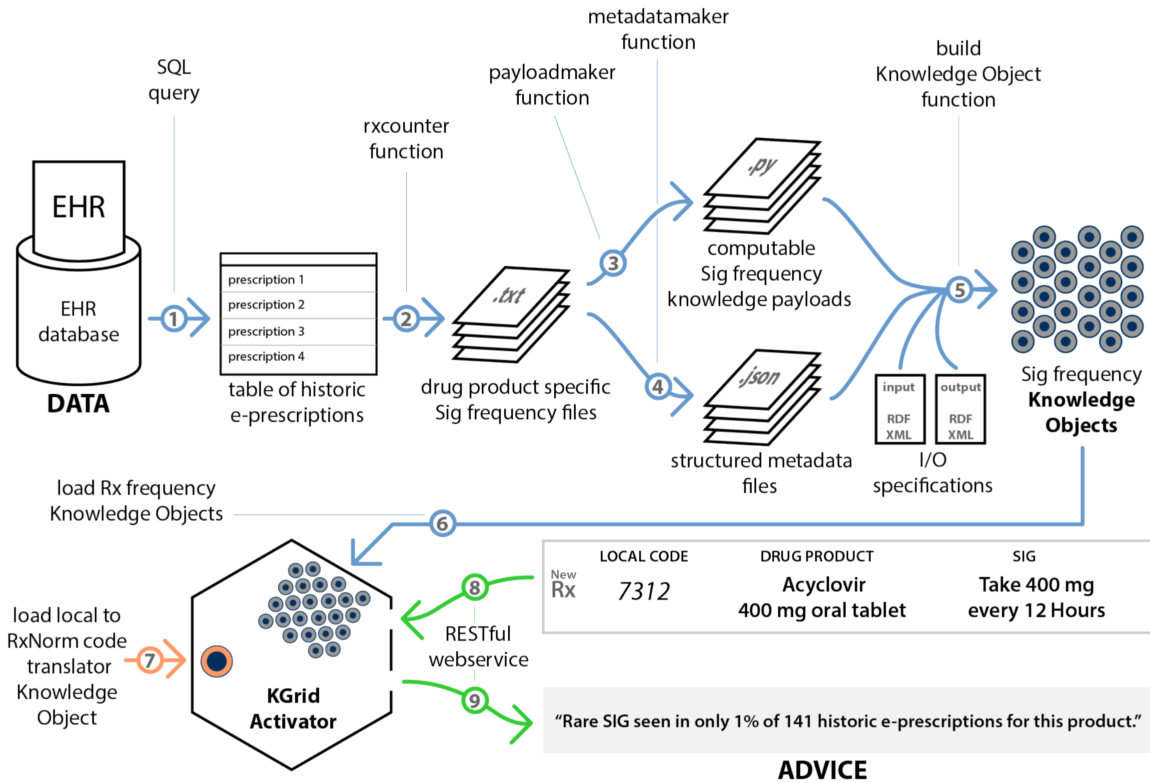


**Figure 4.** ScriptNumerate data-to-advice pipeline showing its components and its 9 essential technical functions

The next two necessary technical functions, indicated by ③ and ④, are the **payloadmaker** and **metadatamaker** functions. These two functions convert Sig frequency text files into executable Python-encoded "payload" files and produce a corresponding structured metadata file in JavaScript Object Notation (JSON), respectively. Each payload is an instance of CBK that will become the core component of a unique Knowledge Object. Similarly, each structured metadata file serves as a framework to organize and describe the multiple components comprising each KO.

The **build Knowledge Object function**, indicated by ⑤, is a pivotal ScriptNumerate function. It automatically unites four digital components: payloads, structured metadata files, an input specification, and an output specification, to form individual *Sig Frequency Knowledge Objects*, each with its own unique identifier. The compound digital Knowledge Objects created at this step conform to KGrid's formal specification of what a Knowledge Object is[14].

The next required technical function, indicated by ⑥, involves **loading Knowledge Objects into the KGrid Activator**. This is a matter of moving digital copies of these Knowledge Objects into a special folder for the Activator.

The seventh technical function, indicated by ⑦ on the lower left of Figure 4, involves a special Knowledge Object to further enhance interoperability. This KO holds a lookup table that maps local medication ID codes from our EHR to corresponding RxNorm SCDs. Thus, this special KO acts as a **code translator**, enabling e-prescriptions carrying only a local medication ID code to be compared with appropriate Sig Frequency KOs in ScriptNumerate.

The last two functions are linked. They are indicated by the green arrows and the numbers ⑧ and ⑨ in Figure 4. These are the functions of ScriptNumerate's API, which is enabled by the KGrid Activator. This API is a RESTful webservice using the WWW architecture. It allows external systems to **send e-prescription Sigs** to ScriptNumerate (⑧) for which ScriptNumerate **computes and returns a message of advice** about each Sig (⑨).

Summarizing, ScriptNumerate has nine required technical functions. Of these, functions ① ② ③ ④ ⑤ and ⑦ in Figure 4 were developed for this study. The KGrid Activator was built beforehand in our lab. It directly supports functions ⑥ ⑧ and ⑨. These nine functions comprise a working data-to-advice pipeline that computes e-prescription advice.

***RQ2.*** *What types of atypical e-prescription Sig advice, and how much of each type, can be generated by using ScriptNumerate to compare contemporary e-prescriptions for an elderly adult population of hospital inpatients with historical e-prescribing patterns arising from treatment of a similar population during the previous year?*

ScriptNumerate computes four different types of advice. First, it advises whether an e-prescription is for an infrequently used or frequently used drug product. Three more advice types arise when e-prescriptions are for frequently used drug products. Then ScriptNumerate advises whether a Sig is common, rare, or unprecedented in comparison to statistical frequencies of past Sigs for the same drug product prescribed for a similar population.

Results from our first ScriptNumerate experiment are reported in Table 3 below. We found that 15,608 (6.2%) out of 251,928 e-prescriptions placed in 2017 were placed for one of 1388 infrequently used drug products. No determination of whether the Sigs for these e-prescriptions were unprecedented, rare, or common could be made. However, in these cases, advice to pharmacists and others could indicate when an infrequently used drug product has been prescribed.

For 431 commonly prescribed drug products, another 14,456 e-prescriptions had Sigs that were unprecedented. Unprecedented Sigs were Sigs not seen during the prior year for a given drug product. In these cases, interruptive alerts advising that a Sig is unprecedented could be warranted to improve e-prescription review. While not every unprecedented Sig suggests a problem, pharmacists are responsible for evaluating whether an e-prescription carrying an unprecedented Sig is warranted or whether, as sometimes happens, the prescriber has made a mistake.

For e-prescriptions placed for the same 431 commonly prescribed drug products, two different thresholds were used to determine whether or not Sigs were "rare." The first threshold defined rare as a Sig appearing in 5% or fewer of historical Sigs for the same drug product. At a 5% threshold, 41,297 of 236,320 e-prescriptions Sigs were considered rare (17.5%). When the threshold for rare Sigs is raised to be less than or equal to 10% of historical Sigs for the same drug product, then 64,443 of 236,320 e-prescription Sigs were found to be rare (27.6%). While interruptive alerts may not be warranted for rare Sigs, tagging e-prescriptions with advice that a Sig is rare could prove helpful in some cases.

**Table 3.** Results from comparing 2017 e-prescriptions using Sig frequencies as historical prescribing pattern data from 2016

| 2017 e-Prescriptions Compared to 2016 Data | For an Infrequently Used Drug Product | For one of 431 Frequently Used Drug Products | | | | |
|---|---|---|---|---|---|---|
| 251,928 | 15,608 (6.2%) | 236,320 (93.8%) | | | | |
| | | Unprecedented Sig | Rare Sig defined at < = 5% Threshold | | Rare Sig defined at < = 10% Threshold | |
| | | | Common | Rare | Common | Rare |
| | | 14,456 (6.1%) | 180,568 (76.4%) | 41,297 (17.5%) | 157,422 (66.7%) | 64,443 (27.3%) |

***RQ3.*** *How can ScriptNumerate aggregate CBK about historical e-prescriptions from multiple EHR sources to form combined collections of Sig Frequency Knowledge Objects?*

One of the potential benefits of using compound digital objects, like Knowledge Objects, to package, manage, and deploy CBK is that components of compound digital objects can be designed to be automatically aggregated. To address ***RQ3***, we added an aggregation capability to the ScriptNumerate data-to-advice pipeline. To do this, we applied the RDF Data Cube vocabulary (www.w3.org/TR/vocab-data-cube/) to represent Sig frequency files as *data cubes*. This vocabulary is based on the Statistical Data and Metadata Exchange (SDMX) International Organization for Standardization (ISO) standard 17369 for statistical data exchange. It enables publishing of multi-dimensional statistical data using RDF triples. This addition to ScriptNumerate is portrayed in Figure 5 below.

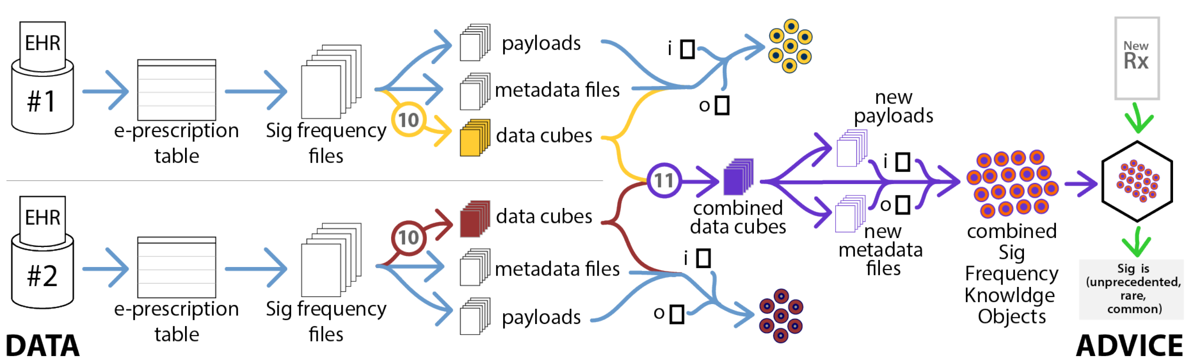

**Figure 5.** How ScriptNumerate aggregates data about e-prescription Sigs from multiple EHRs (i = input spec, o = output spec)

On the left side of Figure 5, the beginnings of two ScriptNumerate pipelines are shown. The top pipeline starts with data from EHR database #1 and the bottom one starts with data from EHR database #2. However, what is illustrated moving from left to right in the middle of Figure 5, starting at ⑪, is a *third combined ScriptNumerate pipeline* built up from aggregating the results in Sig frequency files from the top and bottom ScriptNumerate pipelines.

To both pipelines in Figure 5, a tenth technical function (⑩) has been added to **represent Sig frequency files as data cubes**. The top pipeline's data cubes are in yellow and the bottom pipeline's in red. An eleventh technical function (⑪) **automatically combines the yellow data cubes with the red data cubes to form purple combined data cubes**. These *combined data cubes* are aggregates of the Sig frequency data from both the top and bottom pipelines. From them, new payloads and metadata files can be generated, giving rise to a third collection of combined Sig Frequency Knowledge Objects for computing advice based on CBK from both EHRs. This additional capability to aggregate Sig frequency files represented using the RDF Data Cube vocabulary enables CBK from many sources to be combined.

**Discussion**

This initial attempt to use a single shared compound digital object format to both communicate new CBK and to generate advice by directly applying it was successful. As in Figure 1, our ScriptNumerate data-to-advice pipeline produced compound digital objects holding historical prescribing pattern CBK and then exposed those compound digital objects using an API, but without changing them in any way, thereby enabling ScriptNumerate to generate and communicate advice about new e-prescriptions automatically.

To do this, besides using a formal specification compound digital objects called Knowledge Objects to package CBK, we increased the interoperability of CBK in several other ways: We used strong standards, notably RxNorm, for drug product identification, and also the RDF Data Cube vocabulary, for aggregating CBK. We used several procedural computer languages, including PERL for data transformation and Python to encode prescribing patterns. We employed the KGrid Activator, which provided a specification-based API built using the uniform interface for the WWW. And, because KGrid's technical components run in any suitable Java environment, including cloud environments, we gained a high-degree of mobility for ScriptNumerate and its API.

This work is generally important because more highly interoperable CBK is needed for advice-giving systems to remain current as the volume, velocity and variety of CBK production increase. We expect large increases in these things as 'Big Data' analytics, machine learning, and other computational methods are applied to large data sets to improve health. We believe these increases will result, in part, from automated data analysis pipelines that are fed regularly with new data and rapidly compute results which manifest CBK. Here we have shown that, by increasing the interoperability of CBK in multiple ways, it is possible to extend data analysis pipelines to form data-to-advice pipelines that are capable of both analyzing data to create CBK and providing advice by using the CBK they create.

Our findings from running ScriptNumerate also have important implications for pharmacy practice. They illustrate what could result from deployment of a data-to-advice pipeline for broad atypical e-prescription screening covering hundreds of drug products. In this year-over-year analysis, for an elderly subpopulation of inpatients, we found that 6.2% of e-prescriptions were for infrequently used drug products. We also found that another 6.1% were for commonly used products but had unprecedented Sigs not seen the year before. These results show that ScriptNumerate can highlight atypical e-prescriptions to help pharmacists and others who review them for safety. Therefore, we plan to further develop and then trial ScriptNumerate for multiple subpopulations at one or more sites.

This study has a number of limitations. Perhaps its greatest limitation is that it does not include results from an implementation of ScriptNumerate in practice. In the future, we plan to integrate the KGrid Activator with EHRs and then execute a real-world trial of ScriptNumerate. Another limitation is technical. In its current version, ScriptNumerate requires a Sig contained within a simple JSON text object be the input to its API. It does not yet include desired data pre-processing capabilities to directly parse e-prescriptions formatted using common e-prescription data schemas. We look forward to adding these capabilities to a future version.

**Conclusion**

ScriptNumerate is a working data-to-advice pipeline that increases the interoperability of *computable biomedical knowledge* in several ways. As a prototype, ScriptNumerate demonstrates the potential to support health organizations needing both to generate knowledge from real-world data and then implement the knowledge they generate in practice to provide useful advice. A trial of ScriptNumerate facilitated a year-over-year comparison of e-prescription Sigs for elderly hospital inpatients. We found that 6.1% of e-prescriptions in the second year had unprecedented Sigs. ScriptNumerate's capability to identify atypical e-prescriptions could help organizations improve medication safety.

## References


1. Black AD, Car J, Pagliari C, Anandan C, Cresswell K, Bokun T, et al. The impact of eHealth on the quality and safety of health care: a systematic overview. *PLoS Med* 2011.
2. Lehmann CU, Jaulent MC, Séroussi B. Silver Anniversary: 25 Editions of the IMIA Yearbook. *Yearb Med Inform* 2016; S3–S5.
3. McEvoy D, Gandhi TK, Turchin A, Wright A. Enhancing problem list documentation in electronic health records using two methods: the example of prior splenectomy. *BMJ Qual Saf* 2017.
4. Harrison M, Koppel R, Bar-Lev S. Unintended consequences of information technologies in healthcare - an interactive sociothecnical analysis. *J Am Med Informatics Assoc* 2007; 14: 542–549.
5. Greenes RA, Bates DW, Kawamoto K, Middleton B, Osheroff J, Shahar Y. Clinical Decision Support Models and Frameworks: Seeking to Address Research Issues Underlying Implementation Successes and Failures. *J Biomed Inform*. Epub ahead of print 2017. DOI: 10.1016/j.jbi.2017.12.005.
6. Sittig DF, Wright A, Osheroff J, Middleton B, Teich JM, Ash JS , et al. Grand challenges in clinical decision support. *J Biomed Inform* 2008; 41: 387–92.
7. Kahn R, Wilensky R. A framework for distributed digital object services. *Int J Digit Libr* 2006; 6: 115–123.
8. Chen D, Doumeingts G, Vernadat F. Architectures for enterprise integration and interoperability: Past, present and future. *Comput Ind* 2008; 59: 647–659.
9. De Sompel H Van, Nelson ML. Reminiscing about 15 years of interoperability efforts. *D-Lib Mag* 2015; 21.
10. Paepke A, Chang C, García-Molina H, Winograd T. Interoperability for digital libraries workdwide. *Commun ACM* 1998; 41: 1–23.
11. Bender D, Sartipi K. HL7 FHIR: An Agile and RESTfulApproach to Healthcare Information Exchange. 2015; 1–6.
12. Raths D. What Is CDS Hooks ? An Interview with Josh Mandel About FHIR and Clinical Decision Support. *Healthcare Informatics*, 2016.
13. Mandl KD, Mandel JC, Murphy SN, Bernstam EV, Ramoni RL, Kreda DA, et al. The SMART Platform: Early experience enabling substitutable applications for electronic health records. *J Am Med Informatics Assoc* 2012; 19: 597–603.
14. Flynn A, Boisvert P, Friedman CP, Landis-Lewis Z, Lagoze C. Knowledge Object Reference Ontology. *National Center for Biomedical Ontology Bioportal*, http://bioportal.bioontology.org/ontologies/KORO (2017, accessed 25 January 2018).
15. Flynn A, Shi W, Fischer R, Friedman CP. Digital Knowledge Objects and Digital Knowledge Object Clusters: Unit Holdings in a Learning Health System Knowledge Repository. *Hawaii Int Conf Syst Sci 2016*.
16. Gabriel MH, Swain M. E-Prescribing Trends in the United States. *ONC Data Br No 18*; July.
17. Nelson SJ, Zeng K, Kilbourne J, Powell T, Moore R. Normalized names for clinical drugs: RxNorm at 6 years. *J Am Med Inf Assoc* 2011; 18: 441–448.
18. Flynn AJ. Opportunity cost of pharmacists' nearly universal prospective order review. *Am J Health Syst Pharm* 2009; 66: 668–70.
19. Melton B. Systematic Review of Medical Informatics–Supported Medication Decision Making. *Biomed Inform Insights*; 9. Epub ahead of print 2017. DOI: 10.1177/1178222617697975.
20. O'Donnell J, Vogenberg F. Legal Risk Management Opportunities , Pharmacy Practice , and P & T Committees Part 1 : Deconstructing Dispensing Errors. *Heal Care Law* 2014; 39: 559–563.
21. Woods AD, Mulherin DP, Flynn AJ, Stevenson JG, Zimmerman CR, Chaffee BW. Clinical decision support for atypical orders: detection and warning of atypical medication orders submitted to a computerized provider order entry system. *J Am Med Inf Assoc*. 2014;21(3):569-73.
22. Kane-Gill SL, O'Connor MF, Rothschild JM, Selby NM, McLean B, Bonafide, CP, et al. Technologic Distractions (Part 1): Summary of Approaches to Manage Alert Quantity with Intent to Reduce Alert Fatigue and Suggestions for Alert Fatigue Metrics. *Crit Care Med* 2017; 45: 1481–1488.
23. Ferrández O, Urbina O, Grau S, Mateu-de-Antonio J, Marin-Casino M, Portabella J, et al. Computerized pharmacy surveillance and alert system for drug-related problems. *J Clin Pharm Ther* 2017; 42: 201–208.
24. Friedman CP, Rubin JC, Sullivan KJ. Toward an Information Infrastructure for Global Health Improvement. *IMIA Yearb Med Informatics* 2017; 1–8.
25. Elinder CG, Bárány P, Heimbürger O. The use of estimated glomerular filtration rate for dose adjustment of medications in the elderly. *Drugs and Aging* 2014; 31: 493–499.