APPLYING SOFTWARE ENGINEERING PRINCIPLES TO ELECTRONIC HEALTH RECORDS IN ORDER TO IMPROVE RESEARCH AND GENERATE PATIENT-SPECIFIC ACTIONABLE INFORMATION

A thesis submitted to The University of Manchester for the degree of Doctor of Philosophy (PhD by Published Work) in the Faculty of Biology, Medicine and Health

2020

Richard Williams

School of Health Sciences
Division of Informatics, Imaging and Data Science

Supervisors
Prof. Niels Peek and Dr. Benjamin Brown
Contents

LIST OF TABLES 4
LIST OF FIGURES 5
ABSTRACT 6
DECLARATION 7
COPYRIGHT 9
ELIGIBILITY OF THE CANDIDATE 10
  Degrees 10
  Employment History 10
  Research Experience 10
LIST OF PUBLICATIONS 11
1 INTRODUCTION 12
  1.1 Healthcare data available for secondary use 12
  1.2 Observational research 12
  1.3 Data provenance 15
  1.4 Actionable clinical information 16
  1.5 Software engineering 17
  1.6 Thesis overview and objective 19
2 ORGANISATION OF THE PUBLICATIONS 21
3 CLINICAL CODE SETS 24
  3.1 Background 24
    3.1.1 Clinical codes 24
    3.1.2 Clinical code sets 24
    3.1.3 Sharing code sets 25
    3.1.4 Transparency in code set creation 26
    3.1.5 Code set construction tools and methods 26
  3.2 Term Sets 27
    3.2.1 Term set requirements 28
    3.2.2 Term set definition 28
    3.2.3 Proof of completeness 29
    3.2.4 Code set construction web application 30
  3.3 Existing literature 30
  3.4 Impact of Publications 31
4 MEANINGFUL EVENTS 33
  4.1 Background 33
  4.2 Meaningful Events 35
  4.3 Existing literature 38
  4.4 Impact of Publications 39
5 EVENT SEQUENCES 40
List of Tables

Table 1  The four themes of the thesis.................................................................21  
Table 2  Proportion of reviewed papers performing process mining in each healthcare setting..................................................................................44
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Secondary uses of healthcare data</td>
<td>14</td>
</tr>
<tr>
<td>Figure 2</td>
<td>List of best practices for scientific computing</td>
<td>19</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Relationship of papers submitted for this PhD</td>
<td>23</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Relationship between a term set and a code set</td>
<td>29</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Screenshot of the GetSet web application</td>
<td>30</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Discrepancy between a patient’s true health state and the electronic health record</td>
<td>34</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Mapping of prescription codes to meaningful events</td>
<td>36</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Process mining: Discrepancy between ideal care pathway and actual care pathway</td>
<td>41</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Screenshot from the SMASH web application</td>
<td>50</td>
</tr>
</tbody>
</table>
Abstract

Healthcare services increasingly collect data about their patients in electronic health records (EHRs). These data are also increasingly available for secondary uses, which include, but are not limited to: researchers using the data to perform retrospective observational studies or to recruit to randomised clinical trials; clinicians improving their performance with actionable information from clinical decision support or audit and feedback systems; and national bodies assessing and comparing the quality of care provided across a range of healthcare providers.

The potential benefits of these secondary uses are great, but there is a problem. The data are collected with the primary purpose of direct patient care and therefore caution must be exercised in interpreting, extracting and transforming the data into the form required for the secondary uses. The methods that are used to extract and transform EHR data are of the utmost importance to the subsequent uses; however, they receive much less attention than other aspects of study methodology (such as the statistical analysis) and are chronically underreported in the literature. Without proper reporting and scrutiny of these methods, it is impossible to determine if mistakes or incorrect assumptions have affected the validity of the results. Therefore, confidence in the results is reduced and the impact of the research is diminished.

The viewpoint presented in this thesis is that preparing EHR data for secondary uses is a form of software engineering and therefore should comply with software engineering principles. The objective of the thesis is to present a comprehensive collection of methods and tools, developed in accordance with the best software engineering principles, that bridge the gap between EHR data and secondary uses. These methods include the construction of clinical code sets, the extraction of clinical events from an EHR, and the analysis of sequences of clinical events. In particular, we focus on the use of these methods for observational studies using primary care EHR data, and actionable information as delivered via decision support systems. However many of the methods apply to the full range of secondary uses.

We need robust, open and transparent methodologies, in order to increase the confidence in results from research using EHR data. Only then, can we achieve a much-needed acceleration in healthcare research, and realise the benefits of reproducible, patient-specific, actionable information.
Declaration

Candidate Name: Richard Williams

Faculty: Faculty of Medical and Human Sciences

Thesis Title: Applying Software Engineering Principles to Electronic Health Records in order to Improve Research and Generate Patient-Specific Actionable Information

i. The nature and extent of the candidate’s own contribution and the contribution of co-authors and other collaborators to each of the publications presented is as follows:

Publication 1: Richard Williams conceived the initial idea; performed the research; and wrote the manuscript. All authors: provided guidance as to the scope of the papers; reviewed drafts of the manuscript; and provided comments and proposed alterations.

Publication 2: Richard Williams conceived the initial idea; performed the research; and wrote the manuscript. All authors: provided guidance as to the scope of the papers; reviewed drafts of the manuscript; and provided comments and proposed alterations. Niels Peek coined the name of “term sets”.

Publication 3: Richard Williams conceived the initial idea; performed the research; and wrote the manuscript. Richard Williams and Ben Brown independently rated the performance of the algorithm. All authors: provided guidance as to the scope of the papers; reviewed drafts of the manuscript; and provided comments and proposed alterations.

Publication 4: John Ainsworth conceived the initial idea. Richard Williams performed the research; and wrote the manuscript. All authors: provided guidance as to the scope of the papers; reviewed drafts of the manuscript; and provided comments and proposed alterations.

Publication 5: Richard Williams conceived the initial idea; performed the research; and wrote the manuscript. All authors: provided guidance as to the scope of the papers; reviewed drafts of the manuscript; and provided comments and proposed alterations.

Publication 6: Richard Williams conceived the initial idea; performed the research; and wrote the manuscript. All authors: provided guidance as to
the scope of the papers; reviewed drafts of the manuscript; and provided comments and proposed alterations.

Publication 7: Richard Williams conceived the initial idea; performed the research; and wrote the manuscript. Richard Williams and Colin Davies created the web dashboard. All authors: provided guidance as to the scope of the papers; reviewed drafts of the manuscript; and provided comments and proposed alterations.

Publication 8: Richard Williams conceived the initial idea; performed the research; and wrote the manuscript.

ii. All of the work presented has been completed whilst the candidate has been a member of staff of the University of Manchester.

iii. Publication 4 was submitted as part of the PhD thesis of Professor John Ainsworth at the University of Manchester because he conceived the initial idea and directed the research. It is also included here because, as first author, I performed the research and wrote the paper. No other work in this thesis has been submitted in support of an application for another degree or qualification at this or any other university or other institute of learning.

I confirm that this is a true statement and that, subject to any comments above, the submission is my own original work.

Signed: [Signature] Date: 25/8/2020
Copyright

The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University’s policy on presentation of Theses.
Eligibility of the Candidate

Degrees

2006  PGDip. Business and Management Studies  University of Durham
2003  BA (Hons) Mathematics (1st Class)  University of Cambridge

Employment History

2013 – present  Senior Software Engineer  University of Manchester
2009 – 2013  Software Engineer  University of Manchester
2006 – 2009  Senior Systems Analyst  Vertex Data Science
2004 – 2006  Junior Systems Analyst  Vertex Data Science

Research Experience

Fourteen research papers published in peer reviewed journals up to 2020

Eleven indexed peer reviewed conference papers up to 2020

Presented work at three national, three European and five international conferences

Awarded the John Perry prize for outstanding contribution to primary healthcare informatics by the British Computer Society in 2017

Supervisor of a computer science MSc project

Awarded £2000 for a NIHR short term placement award
List of Publications


1 Introduction

1.1 Healthcare data available for secondary use

Every day in the UK there are over 1 million primary care consultations [1]. When expanded to include all interactions with primary, secondary and tertiary care systems worldwide, there are likely 100s of millions of daily healthcare encounters. These encounters generate large volumes of data [2,3]. These data, whether handwritten, captured electronically as free text, or coded against pre-defined clinical terminologies, are used primarily for patient care. However, there are also many secondary uses, such as researchers using databases of health data to carry out retrospective observational studies [4,5], or clinicians improving their care via actionable information [6–8].

The amount of data captured electronically and available for research is ever-increasing, partly due to the increased capture and computerisation of electronic health records (EHRs) [9,10], and partly due to movements to make these data more readily available for secondary use. Further factors for the increase include a growing population, and an increasingly ageing population [11], whose usage of the health and social care system is disproportionally large compared with younger age groups.

1.2 Observational research

The randomised control trial (RCT) is considered the gold standard for assessing the efficacy and effectiveness of interventions [12]. However, RCTs can be costly both financially and in terms of the time taken to recruit participants and then perform the study. There are also inherent biases in RCTs where certain high-risk populations, such as those with comorbidities, are understandably excluded [13]. This has the potential to affect the generalizability of any results [14].

In 2001, an editorial in the BMJ highlighted that [15]:

12
“...we still do not know which treatments are useful for acute stroke, but if every patient in the world experiencing a stroke were admitted to trials we would have enough patients within 24 hours to answer many of these questions.”

It may be possible to embed such pragmatic point-of-care trials within healthcare systems in the future, so that patients presenting with a condition for which there are multiple therapies are randomly assigned a treatment. At present, aside from a few small scale trials [16,17], this is not possible. Instead, researchers can use the large volume of routinely-collected EHR data to perform retrospective observational trials. Observational studies should never replace RCTs, but they can answer many otherwise impossible questions [18]:

- What is the prevalence of a condition, or a combination of conditions in the real world?
- What is the care that people, with a certain condition, receive (in the real world), and how does this relate to quality standards?
- What is the effectiveness of an (already proven efficacious) intervention in the real world?

Healthcare data are mostly collected for direct patient care. There are a few minor exceptions to this, such as data collected for payments via quality assurance and incentivisation schemes such as the Quality Outcomes Framework (QOF) in the UK [19]. However, data are rarely collected for, or tailored towards, research. Despite this, they remain an attractive resource for researchers. The data are available in large quantities and, besides the barriers concerning information governance when attempting to access potentially identifiable data, easily obtainable. It represents the real world, has a low cost, large population sizes, and long follow-up times [18]. Compare this with a standard research study that must recruit participants and collect data, often over many years and at great expense, before analysis can even take place.
See Figure 1 for a list of secondary uses of healthcare data as identified by a recent literature review [20].

One of the largest limitations of observational studies is that of reproducibility. Goodman et al. define three terms for discussing research reproducibility: methods reproducibility, results reproducibility and inferential reproducibility [21]. Methods reproducibility is the degree to which a publication includes sufficient information such that other researchers could repeat the analysis. Results reproducibility is the degree to which other researchers can achieve the same results. Inferential reproducibility is the degree to which different researchers would reach the same conclusion based on similar results. Most of this thesis focusses on ways to improve methods reproducibility.

The combination of a lack of methods reproducibility, and forms of bias such as selection bias, confounding bias, and publication bias, means that findings in one dataset are often unrepeatable in other datasets [22,23]. This is not unique to observational research. However, if these issues could be resolved, observational
research would be stronger, and its contribution to meta-analyses and clinical guidelines would increase.

1.3 Data provenance

In traditional health research, such as RCTs, the data are captured by a team member or research nurse, following a well-defined study protocol. The provenance is clear as the data have been captured in a pre-defined format for analysis. However, when performing observational studies on EHR data, it is necessary to understand how and why the data were collected. Otherwise, it is hard to view results from the research with confidence. For instance, in UK primary care virtually all prescribing is electronic, therefore the prescription data can be treated as accurate and complete, even if we do not know whether the patient took the medication. Similarly, the automation involved in laboratory test reporting leads to confidence in their recording. Symptoms are typically poorly recorded as coded information, appearing as free text, or omitted altogether if considered unimportant [24–26]. Studies that rely on poorly coded information, without reporting it or taking steps in the analysis to mitigate its effects, will be of low quality and limited use. This concerns policymakers, clinical guideline creators, and authors of systematic reviews, who at best will identify that the studies are of low quality and therefore weight their findings accordingly. However, in the worst-case scenario, these results might form the basis for clinical guidelines that enter practice despite the inferior quality of the evidence behind them, potentially undermining the evidence-based healthcare environment.

When using routinely-collected data, researchers must understand, clean and manipulate the data into a form ready for analysis [27–31]. For most researchers, these extract, transform, and load (ETL) activities take, or should take if done properly, a significant amount of time. However, the bulk of the methods described in academic papers focus on the analysis of the data obtained from the ETL phase, rather than the ETL methods themselves. The ETL methods and the data analysis are equally important, when interpreting and scrutinising the findings, as any errors or incorrect assumptions can alter the conclusions.
Van den Broeck et al. [32] called for a data-cleaning plan in study protocols, which was provided by the “Reporting of Studies Conducted using Observational Routinely-collected Health Data” (RECORD) statement [33]. It suggests that researchers should include information on their data cleaning methods, and their methods of selecting their study population. However, even when this is described, unless it is accompanied by computer code, it can still be hard to reproduce. Without giving information on this critical part of research, it is very difficult for other researchers to reproduce the work with confidence.

1.4 Actionable clinical information

Besides observational research, EHR data can be processed, analysed and delivered to clinicians as actionable information [34]. There are two principal areas, clinical decision support (CDS) [6,7] and electronic audit and feedback (e-A&F) [8]. CDS systems provide support to clinicians for individual patients “at the point of care”, such as symptom checkers, prescription contraindication alerts and diagnostic models. This differs from e-A&F systems that typically present aggregate information to clinicians to allow them to compare their performance with their peers or the past. A major challenge for e-A&F systems is that population-level targets do not easily translate to patient-level actions [2]. More recently e-A&F systems have incorporated patient-level data, supporting clinicians to take specific actions for individual patients that will contribute to population-level targets [35].

Making use of patient data for these systems has one major difference from the observational research use case. With observational research, it is difficult to know whether any results generated are correct or if mistakes have been made. However, systems such as e-A&F dashboards are used by clinicians, who can compare the information with their knowledge of their patients and their clinical systems to determine if what is presented to them tallies with their expectations. Any errors or inconsistencies in the feedback can be identified and fed back, leading to iterative cycles of improvement. However, if not done well, clinicians may stop using the system because of its unreliability. It is therefore still important to systemize the ETL approach, as systems with fewer errors are more likely to be adopted.
1.5 Software engineering

Laplante [36] describes software engineering as:

“…a systematic approach to the analysis, design, assessment, implementation, test, maintenance, and reengineering of software, that is, the application of engineering to software.”

Developing tools and methods with a systematic approach leads to software that is usable, reliable and fit for purpose. However, there is no single set of universally accepted principles of software engineering. Wilson et al. propose a list of 24 best practices developed for the conjunction of research and software [37], but various other authors define 7 [38], 15 [39] and 201 [40] core principles. It is therefore useful to consider two aspects of software engineering. They are the principles of software design, and best practices for software development.

Software design relates to the format and structure of the actual code or packages of code. Common software design principles would include abstraction, modularization, encapsulation and separation of concerns [41]. Abstraction is the identification of the essential components of a piece of software while ignoring the details of the implementation. Encapsulation describes the bundling together of common components into a single package of code. Modularization is the splitting of code into separate components or modules. Separation of concerns describes how modularization should occur; each module should be responsible for a single thing.

Software development focusses on the processes involved in producing software. A list of best practices for this would include version control [42,43], build processes [44,45], testing [46] and issue tracking [47]. Software design principles for well-structured code are universal, whereas some best practices for software development are domain-specific. For example, effective team working [48,49] is important, but in a research setting with limited resources often impossible. Another example is when
code is made open-source, and therefore publically available [50,51]. This might be unwise for a company protecting their intellectual property, but for researchers is an important step in ensuring their methods are reproducible and amenable to validation.

Well-structured code is very important. However, for this thesis, I am not focussing on that area. Instead, when referring to software engineering principles, I am describing the application of the design principles to entire software packages, and the best practices involved in their development. A list, modified from Wilson et al. [37], against which I will assess my work, is shown in Figure 2. Items from the original list were omitted if they were overly specific to routine computing (“Make [variable] names consistent, distinctive, and meaningful”, “Save recent commands in a file for re-use”), or non-specific if it would be hard to assess my work against them (“Make code style and formatting consistent”).
Figure 2 - List of best practices for software engineering in scientific computing adapted from Wilson et al. [37]

1.6 Thesis overview and objective

There is a lack of methodologies for converting routinely-collected primary care data into a form that can then be used for secondary purposes. Those methods that do exist have not always been developed systematically in line with the principles of software engineering.

My objective in this thesis is to address these issues by presenting transparent and reproducible methods for the management of healthcare data. These data can then be used both for further research openly and sustainably, and for actionable information that can be presented to clinicians and used for clinical decision making at the point
of care, thus improving patients' outcomes and their quality of life. Although I focus on observational studies using EHR data, and actionable information as delivered via e-A&F systems, many of the methods apply to the full range of secondary uses.

The following chapter explains the organisation of the papers, their relationships and the four themes that have emerged. Chapters 3-6 expand on each of these four themes, before the discussion and conclusion. Finally, there is an epilogue of my publication for the 2018 Christmas edition of the BMJ. This is not included in the body of the thesis as it lacks the scientific rigour of the other papers. However, it highlights some important challenges in the world of clinical code terminologies, and does so in a hopefully entertaining way.
2 Organisation of the Publications

The publications submitted for this PhD encompass some of the steps required to take data collected routinely for patient care, process it so it may be used for research, and analyse it so it can be fed back to clinicians in a way that enables action. The publications are grouped into four themes (see Table 1), which are explored further in the following four chapters.

Table 1 - The four themes of the thesis, and the publications drawn upon for each

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Theme</th>
<th>Publications</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Clinical code sets</td>
<td>1, 2</td>
</tr>
<tr>
<td>4</td>
<td>Meaningful events</td>
<td>1, 3, 6</td>
</tr>
<tr>
<td>5</td>
<td>Event sequences</td>
<td>4, 5, 6</td>
</tr>
<tr>
<td>6</td>
<td>Actionable information</td>
<td>7</td>
</tr>
</tbody>
</table>

The first theme concerns the creation of sets of clinical codes that identify patients and relevant events in a dataset of routinely-collected healthcare data. We must ensure that the code sets are accurate, verifiable, reproducible and reusable. Publication 1 is a review on clinical code set engineering methods and shows the current best practices used by researchers, but also highlights the deficiencies in their approaches. These include the absence of sharing which leads to a massive duplication of effort, and the lack of reporting of how code sets were defined and created which leads to a lack of trust in the results. Publication 2 introduces a better way of creating code sets via a mechanism, which we have named “term sets”. I summarise my work on clinical code sets in chapter 3 and draws on publications 1 and 2.

The second theme is on meaningful events. We can use code sets (or term sets) to extract data from the EHR. However, the events recorded with clinical codes do not necessarily correspond to real-life events and they require translation. For example, in publication 3, we demonstrated that for medications, the events in a record correspond to the prescription events. However, the events of interest to a researcher (or the user of an e-A&F application) are when the drug was started, when it was stopped, and when the dose was changed. We present a new algorithm that successfully translates...
the raw events in the record to more meaningful events. Chapter 4 describes this work and draws on publications 1, 3 and 6.

Events in isolation are typically of limited interest, so instead, we construct sequences of events that represent a patient's care pathway. Publication 4 demonstrated a novel method for analysing care pathways using string metrics. More often, the analysis of sequences of events is done with process mining, but in publication 5, a literature review examining the setting for process mining research in healthcare, we found that research for these techniques in primary care data was absent, possibly because of the mismatch between events direct from the record and those with meaning. By extracting these meaningful events using our methods in publication 3, we could apply process mining to primary care data successfully in publication 6. I cover this theme of event sequences in chapter 5.

All the above not only accelerates reproducible research but also enables us to develop software that presents actionable information to clinicians so they can improve patient care. The final theme is therefore on actionable feedback and draws from the work of the previous three themes. Publication 7 shows the development of the web application SMASH, which identifies patients at risk of hazardous prescribing and presents that information to a GP or pharmacist who then takes steps to remove the risk to the patient. I discuss the final translation of information into an actionable tool in chapter 6.

The connections between the papers are presented in Figure 3 and this shows the high degree of dependency between the seven papers submitted for this thesis.
Figure 3 - The dependency relationship between each of the papers submitted for this PhD. An arrow indicates that the target paper node was reliant on the research in the source paper node.
3 Clinical code sets

This chapter draws from publications 1 and 2, in the list of publications:

- Clinical code set engineering for reusing EHR data for research: A review.
- Term sets: A transparent and reproducible representation of clinical code sets.

3.1 Background

3.1.1 Clinical codes

Clinical codes are alphanumeric strings of characters that represent clinical concepts such as diagnoses, procedures, observations, medications, and laboratory tests. In UK primary care the coding system, also known as a code terminology or dictionary, that has been used for the last 20 years is called “Read codes” (named after Dr James Read, the physician who created them) [52]. For example, a GP would use the code “G30..” to record an acute myocardial infarction. These codes reduce the amount of free text in a patient’s clinical record, while increasing the ability for the record to be searched and queried electronically. Other common terminologies in use around the world include the International Classification of Diseases (ICD) [53] and SNOMED CT [54]. In the UK, the NHS is migrating all clinical systems to use SNOMED CT [55].

With the exceptions of text mining and natural language processing, the vast majority of computable interactions with healthcare data, both in practice and for research, are via clinical codes. A few examples of the diverse range of queries would be: a researcher defining a cohort of patients for an observational study; a pharmacist identifying patients at risk from hazardous prescriptions; a GP providing direct patient care; and a practice manager producing a list of patients with type 2 diabetes to send out annual review reminders.

3.1.2 Clinical code sets

A single code is rarely sufficient to identify the concept of interest. Instead, a set of codes, or multiple sets of codes, must be created. Our review of clinical code set construction (Publication 1 in this thesis) recommends calling these objects “clinical
code sets”, or “code sets” for short, to ensure researchers speak a common language. Other terms in use include “clinical code lists”, “code lists”, and “value sets”. In complying with our recommendation, we will refer to them from here on as “code sets”.

Let us consider the example where we are identifying patients with type 2 diabetes mellitus (T2DM). We would first find patients with a diagnosis of T2DM. It varies by terminology but there are likely 10s or 100s of diagnosis codes for T2DM, each with a different nuance. Clinicians only prescribe Metformin to patients with T2DM, so a patient with an active prescription for Metformin likely has T2DM, even without a diagnosis code. We would therefore compile a code set for all variants of Metformin. Finally, a patient’s HbA1c level (blood glucose) can indicate whether a patient has T2DM. Several high readings might indicate that a patient has T2DM even if a clinician has not yet spotted it or recorded a diagnosis in the record, so we would want to create a code set for HbA1c. Depending on the exact reason for finding patients with T2DM, we may use some or all of these code sets.

3.1.3 Sharing code sets

It is important that code sets are shared and findable. Without sharing there is a duplication of effort, as the same code sets are developed by different groups of researchers. Researchers have highlighted the deficiencies of the existing sharing paradigm. A review by Springate et al. [56] found that in 2014 only 5% of publications using code sets made them available to other researchers. Even when made available as an appendix or in supplementary material, arguably, the code sets are not as findable as if they were published in a dedicated repository [57,58]. Our paper on term sets (Publication 2 in this thesis) indicates that this has improved with 44% of papers reviewed making their code sets available for reuse. Our literature review into the methods of code set construction (Publication 1) found 14 publications calling for code set sharing. Gulliford et al. say there is “a need for greater transparency in the selection of sets of codes for different conditions,... as well as sharing of code sets among researchers” [59], Winnenburg and Bodenreider call for “an authoritative source of value sets” [60], and Bhattari et al. highlight “the need for transparency in the reporting of case definitions” [61].
Why are code sets not routinely shared? Rañopa et al. [62] suggest that the authors do not consider the code sets as an important part of their research and so omit them on the grounds of a lack of interest. Alternatively, researchers who have invested a lot of time in producing code sets may want to protect their interests by not giving them away for free. Benchimol et al. [33] also highlight this but argue that unless protected by law or contract, code sets should be made available under the scientific principle of transparency.

3.1.4 Transparency in code set creation

The method used to construct code sets must be transparent and publically available. Without this, even code sets that are available and findable will see limited reuse because of a lack of confidence in their quality. A code set could be downloaded and naively used without checking, but to confirm that it is of acceptable quality a researcher must check it. This involves inspecting each code in the set to determine if they apply to the study, before exhaustively searching the entire code dictionary for codes, which may have been omitted. When faced with this checking task, which will take a similar amount of time as construction from scratch, it is clear why this lack of metadata is a barrier to the reuse of code sets. Finally, codes that are not present in the code set may have been omitted accidentally because of deficiencies in the construction process, or they may have been deliberately excluded for study-specific clinical reasons. Without the metadata showing how the code set was constructed this decision, or mistake is unknowable.

Other researchers have also identified this problem, with Defalco et al. [63] discussing the benefits of the “inclusion of... code sets and a description of the process used to develop code sets in publications”, and Nicholson et al. [64] proposing that “the publication of precise search terms and results, may allow transparency and replicability in the preparation of code-lists and facilitate their sharing”.

3.1.5 Code set construction tools and methods

There are several tools and methodologies for creating code sets in a way that is open to scrutiny. Davé and Petersen’s approach involves creating a code set via a Stata script [65]. The script allows researchers to search for synonyms of the concept of interest,
and makes use of the hierarchy of the terminology to find the correct codes. The Stata script can be shared to demonstrate how the code set was constructed.

Olier et al. [66] created an R package (Rpcdsearch) and a Stata command (pcdsearch). As with the previous method, these allow researchers to create code sets by combining synonym searching and hierarchy traversal, but with the addition of Boolean operators in the search. The Clinical Research using Linked Bespoke Studies and Electronic Health Records (CALIBER) [67] project also has an R package (CALIBERcodelists) [67]. This adds the ability to search with regular expressions while maintaining the same features as the two previously mentioned approaches. Both methods could be made available for scrutiny by sharing the R scripts used for the code set generation, though the authors do not discuss this.

Watson et al. [68] described a 3-stage process. Part of this process is a Stata script similar to earlier approaches. The novelty is an additional stage, where domain experts and clinicians undertake a Delphi exercise to reach consensus on the codes included in the set.

The existing approaches share certain strengths; however, they also have their limitations. The methods are tied to either R or Stata, with the metadata being outputted in a script. An approach independent of the scripting language would be better. For the methods that allow terms to be excluded, the exclusion is achieved by specifying the particular code from the terminology. This leads to code sets that are not generalizable to other terminologies.

3.2 Term Sets

The limitations described in the previous section showed that there was the opportunity to develop a new representation of clinical codes sets, and a method for their creation. For both the new representation and the new method, several prerequisites needed to be met to ensure that they were fit for purpose and that researchers would adopt them in preference to their existing approaches.
3.2.1 Term set requirements

For the new representation of code sets, it should contain sufficient metadata to allow the construction of the code set to be scrutinised, replicated and re-used. If the steps taken to build the code set were available transparently, then researchers could assess for themselves whether the code set was suitable for their purposes. They could also make minor adjustments if required without having to start from scratch, thus avoiding the duplication of effort described earlier. If the method of construction was described in a computer-readable way, then software could be developed to assess and validate code sets automatically against several metrics.

For the new method to create the code set and the associated metadata, it should be comparable with previous methods at finding all applicable codes. Researchers will not use a tool in preference to others unless it is superior. It should also be as easy, if not easier, than existing methods in use. If the method were better than existing ones, but also complicated or time-consuming to use, then the incentive for other researchers to adopt it would be limited. However, if it was superior, and also easier and quicker to use, then the only challenge to adoption would be in raising awareness within the community.

3.2.2 Term set definition

We therefore created “term sets” which comprise three parts: a list of inclusion terms describing the feature of interest (e.g. ‘myocardial infarction’, ‘chest pain’); a list of exclusion terms to remove codes of no interest (e.g. ‘family history of’, ‘myocardial infarction ruled out’); and the terminology and version (e.g. terminology = Read code, version = v2-20180401). The code set is then uniquely determined by searching the target terminology with the list of inclusion terms while rejecting anything that matches an exclusion term.

Exclusion terms provide an important level of metadata that is not currently present in code sets. They allow researchers to see whether certain codes have been excluded accidentally or deliberately. See Figure 4 for the relationship between a term set and a code set.
Figure 4 - Relationship between a term set and a clinical code set. The clinical code set can be uniquely constructed from the term set. In this case, the term set for type 2 diabetes mellitus can generate a code set for the Read v2 terminology.

### 3.2.3 Proof of completeness

I demonstrated that a term set could represent any code set with a formal proof. This was important to provide assurances that term sets could be adopted without the risk that it may be impossible to represent a particular code set.

It is perhaps obvious that lists of inclusion and exclusion terms are easier for researchers to read and understand when compared with a list of codes and their definitions, but we needed to go further. We therefore performed a literature search to find papers with code sets. I converted these code sets to their equivalent term set representation and compared the sizes. The results were that term sets were on average 74% the size of their equivalent code set, with only 4 term sets larger than their code set. However, this was in part due to deficiencies in the original code sets, which when corrected saw all term sets being shorter than their code sets.
3.2.4 Code set construction web application

I developed the GetSet web application (https://getset.ga) as a simple way for researchers to manage code sets – see Figure 5. The site allows: new code sets to be created; existing code sets to be validated against several metrics; existing code sets to be shared publically; existing code sets to be copied and modified for different purposes.

Figure 5 - Screenshot of the home screen of GetSet. GetSet is a web application for the creation, validation, and sharing of clinical code sets.

3.3 Existing literature

SNOMED CT allows users to create Reference Sets [69] which are a way to define a subset of the code hierarchy. While this is a useful feature, there is still the need for the methodology outlined above to assist the researcher to create the intensional Reference Set, in a way that facilitates transparency and reuse. Reference sets are also specific to SNOMED CT whereas term sets, and the methodology and software tool for their creation, are generic and not terminology-specific.

There is also a Health Level Seven (HL7) Fast Health Interoperability Resources (FHIR) concept of a ValueSet [70] for defining value sets, or as I would call them, code sets. The ValueSet can be extensional or intensional. Intensional sets comprise a
set of rules and Boolean operators that generate a code set when applied to a particular version of a terminology. As with the Reference Sets above, this is a way to represent code sets but does not provide any support for their creation. A future improvement to the GetSet tool would be the ability to export term sets as intensional FHIR value sets leading to an open process for code set creation, coupled with the precision of the FHIR definition.

Chu et al. [71] compared intensional with extensional code sets in SNOMED CT. They looked at 10 clinical conditions and compared the conciseness of the extensional and intensional definitions, as well as the length of time taken to curate them. They found that the intensional definitions were more concise (median 3 versus 78 concepts to define) and quicker to build (5 versus 37 minutes). This confirms our findings from the term set publication.

3.4 Impact of Publications

“Clinical code set engineering for reusing EHR data for research: A review” has been cited 29 times since publication. Julie George of the Population Health Analytics Network said it was “an excellent recent systematic review on methodologies used to develop code lists”. Peter Embi selected it to include in his Clinical Research Informatics (CRI) Year-In-Review session presented at the 2018 AMIA Informatics Summit [72]. Peter reviews all papers on CRI over the previous year and selects those he considers to have contributed notably to the field. ResearchGate calculates a Research Interest score for each publication. This paper has a score of 15.9, which is higher than 86% of all items on the site, and higher than 95% of those published in 2017.

“Term sets: A transparent and reproducible representation of clinical code sets” was recently published and has been cited twice. An abstract of the paper was presented at MedInfo 2019, the leading international conference for health informatics. The web application described in this paper has seen moderate traffic. The ResearchGate Research Interest score is 2.4 which is higher than 73% of papers published in 2019.

Reviewer comments for term set paper:
1. “I found this very interesting and in many ways quite overdue. I can completely see the value of term sets and will endeavour to use them in my own future work”

2. “This is a well written interesting paper. The introduction gives a clear description of the current problems with clinical codelist development. Term set is a neat way of defining the inclusion and exclusion criteria for a clinical codelist, similar to previous proposed methods, but independent of any particular statistical software.”
4 Meaningful events

This chapter draws from publications 1, 3 and 6, in the list of publications:

- Clinical code set engineering for reusing EHR data for research: A review.
- Making medication data meaningful: Illustrated with hypertension.
- Process Mining in Primary Care: Avoiding Adverse Events Due to Hazardous Prescribing.

4.1 Background

The clinical codes in a patient’s record provide a summary of what has happened to that patient during interactions with the healthcare system. They can be considered as an event log. Examples would include: the patient is diagnosed with hypertension; the patient receives metformin for controlling their diabetes; or the patient is admitted to hospital. However, this is only a filtered view of the events that have occurred (see Figure 6).

There are problems with completeness, accuracy and bias [3]. For measurements and observations, the record contains infrequent snapshots. Not all symptoms are remembered or reported by the patient. Those that are reported may not be considered relevant by the clinician and so not entered into the record, and those that are entered may be done so as free text and not coded [25]. We know when medicine is prescribed, but we do not know whether it is taken according to instructions [73]. Diagnoses typically occur several months or years after the actual condition first occurred [74–76]. Exercise [77] and dietary information [78] are only captured in response to clinician questioning and are subject to recall bias and patient honesty. Finally, anything entered into the patient’s record during a consultation would be timestamped as if it had occurred on that day, whereas in practice these events would have occurred before the consultation.

The EHR is a filtered and obfuscated view of the patient’s actual condition. All research using coded data from EHRs needs to be aware of this limitation and use strategies to mitigate the associated risks.
Figure 6 – The discrepancy between a patient’s true health state and the electronic health record

Usage is also a problem. Clinicians in UK primary care sometimes enter a diagnosis code when a review is taking place. This is not a problem for chronic conditions such as diabetes or hypertension. However, for acute conditions such as stroke or myocardial infarction, which are discrete events requiring review, it is difficult to determine if the patient has experienced another event, or if a clinician reviewed their pre-existing condition.

The events in a record carry a significance over and above their apparent meaning. A patient’s HbA1c level is important when deciding if a patient is developing type 2 diabetes. However, that the clinician ordered the test is significant as it shows they already suspected the diagnosis. A study has shown that the presence of most laboratory tests, irrespective of the results, has a significant association with survival [79]. The same study found that in 118 out of 174 tests analyzed, the time of day that the tests were requested was more accurate than the results of the test itself at predicting survival. This is because tests for healthy people are rarely requested at
weekends or in the middle of the night. Other studies have shown that the frequency of measurement, again irrespective of the results, can improve predictive models [80,81]. Another example would be a clinician requesting a diagnostic test to check for something common before starting an investigation for something more serious. Here, a normal value for the test result would show that there might be a more serious problem, while an abnormal value would not. Finally, codes in the record can indicate that nothing has happened. For example, consider a repeat prescription where a medication is reissued as a continuation of a previous prescription. The patient’s record contains a code showing that the drug was prescribed, which might be of interest. However, the continuation of the medication shows that the patient’s condition is unchanged and that no clinically meaningful event has occurred.

For some of these issues, it is not possible to take corrective action, and we can only remain vigilant to the limitations of routinely-collected data and design our studies to be robust in the face of these limitations. However, for many of these issues, it is possible to mitigate them by creating new events with more meaning that can be added to the dataset to be available for future processing.

4.2 Meaningful Events

Meaningful events are things that happen to patients, but they do not necessarily correspond one-to-one with events in the EHR. Publication 3 describes the method and algorithm that I developed to take prescription events coded in an EHR and extract the therapeutic decisions made by clinicians. In UK primary care, where prescribing is almost entirely electronic, a patient’s EHR contains a log of all the prescriptions given to the patient. However, the prescription events alone are not as meaningful as the therapeutic decisions made by clinicians, such as when a drug is started, stopped and when the dose is changed (see Figure 7).
Figure 7 - Medication prescription codes in an EHR need converting into meaningful clinical events such as when medication was started or stopped.

To calculate the drug start, stop and dose change events we need four pieces of information: the date of prescription, the active ingredient, how much was dispensed, and the rate of consumption. The algorithm takes the data from an EHR and first determines if enough information is present to continue. Different EHRs capture information in different ways, but the algorithm adapts accordingly. For example, a clinical code can uniquely determine the active ingredient or ingredients, the form of the medication such as tablet or liquid, and the amount of active ingredient per unit quantity of medication. The algorithm will use either the clinical code or the three pieces of information depending on what is present. Another example is the instruction given to the patient, such as “take 2 with mealtimes”. With this, we can determine the rate of consumption via regular expressions, if that information is not otherwise available. Finally, the algorithm allows certain parameters to be changed, such as the length of time after a prescription has run out before their next prescription commences before deciding that the patient has stopped the medication, rather than their adherence has reduced.

In datasets with linked pharmacy and GP data, it would be possible to determine whether a pharmacist dispensed a prescription created by the GP. In in-patient settings, it may be possible to go further and observe a patient taking medication to record their
compliance with the instructions. However, in GP data alone, these steps are not possible. The best we can infer is whether a patient has sufficient medication. A patient who receives a prescription lasting 28 days, who then gets their next prescription more than 28 days later, has not received enough medication to adhere to their prescription.

Publication 6 in this thesis describes our application of process mining techniques to routinely-collected primary care data. This had not been previously performed or reported on meaningfully. The next chapter provides a more detailed view of process mining and event sequences. However, it is mentioned here, because we translated codes in a patient record into events that would be part of their care process. The absence of research in this area prior to our publication may be due to a lack of recognition of this extra step that must be performed.

In UK primary care, diagnoses are rarely made from a single observation. Let us consider the diagnosis of chronic kidney disease (CKD) from the current NICE clinical summary [82]. When a clinician suspects CKD, they measure the patient’s estimated glomerular filtration rate (eGFR). If it is below 60 mL/min/1.73m$^2$ then they repeat the test within two weeks. If it remains below 60, then they order a further test within 3 months. If their eGFR is still below 60, showing a persistent reduction in kidney function, then the clinician diagnoses CKD. Therefore, an eGFR of 50 in a patient’s record can mean at least four different things: the patient is suspected of having CKD and a follow-up in 2 weeks should be ordered; a patient requires a follow-up test in 3 months; a patient should be diagnosed with CKD; or a patient with CKD has had their CKD routinely measured. The context of the clinical record must be used to create events that are more meaningful.

There is an unavoidable degree of bias introduced when constructing meaningful events. When several high systolic blood pressure measurements appear in a patient’s record, should there be a meaningful event? If so, is it that multiple blood pressures were measured, is it that a second test was performed after an earlier reading, or is it that 2, 3 or 4 high readings occurred within a certain amount of time? The meaningful events are likely correlated which may mitigate this bias. A better way is to ensure that we use clinical guidelines. Defining meaningful events in a way that corresponds to
the advice given in guidelines will allow us to take information in a record and map it more closely to the decisions made by clinicians, which is the ultimate aim.

4.3 Existing literature

There is very little literature about meaningful events or similar concepts. The closest area of research is electronic phenotyping, which aims to discover patient phenotypes from their EHR. Patients are classified into groups such as *patients with heart failure*, or *patients with a history of depression who have insomnia and are receiving antidepressants*. This is done via rule-based database queries, natural language processing and machine learning [83,84].

In the literature review on clinical codes, I reviewed all publications relating to electronic phenotyping projects. This was to discover how they managed the creation, curation and sharing of code sets required to construct the phenotypes. However, it is worth reflecting on the different approaches that different platforms adopt. At one extreme there are systems where the creators define the phenotypes in advance; the end-user has no control over their definition and simply requests all patients with, for example, the ‘hypertension’ phenotype [85–87]. The other extreme provides end-users with the flexibility to construct their phenotypes in any way they like, even using bespoke clinical terminologies with no need to map to a standard dictionary [88,89].

Mannhardt et al. [90] talk about the need to create an abstraction log from the underlying event log. Where we have used the term meaningful events, they use the term “activity”, but they are synonymous. However, their mapping was from a clinical system in secondary care that recorded very granular information, which was meaningless unless grouped together, and their supposed generic approach was, in fact, specific to the example where a single activity results in several events. Their observation that the abstracted events “*are more useful in the communication with stakeholders, since they refer to recognizable activities*” is sound and important.

Regardless of the approach, there is still inherent bias introduced when a phenotype or meaningful event is defined. However, provided the definitions are created in line with current best practice and clinical guidelines, and that they are made available for scrutiny, then the potential problems arising from this are mitigated.
4.4 Impact of Publications

The impact of publication 1 was described in the previous chapter.

“Making medication data meaningful: Illustrated with hypertension” has been cited five times since publication. It was presented at Medical Informatics Europe (MIE) 2016, the leading European informatics conference, where it received the top classification of “strong acceptance”. Reviewers’ comments included:

- “This is an important technique to make clinical sense of EHR and should be definitely included in the conference program.”
- “Well structured and interesting work!”
- “Well written paper, as a methodology paper with an interesting example to illustrate the method”

“Process Mining in Primary Care: Avoiding Adverse Events Due to Hazardous Prescribing” was published recently and has not been cited. The paper was presented at MedInfo 2019, the leading international conference for health informatics. Reviewer comments included:

- “Very interesting analysis of a very large data set. Also very important its focus on primary care.”
- “Good literature review”
- “The work seems well performed, the results and their interpretation useful. The discussion contains relevant observations and lessons taken, and adds value to the paper.”
- “The paper is very clearly written overall”
5 Event sequences

This chapter draws from publications 4, 5 and 6, in the list of publications:

- Using String Metrics to Identify Patient Journeys through Care Pathways.
- Process mining in primary care: A literature review.
- Process Mining in Primary Care: Avoiding Adverse Events Due to Hazardous Prescribing.

5.1 Background

When viewed in isolation, clinical events, even the meaningful events described in the previous chapter, are not usually sufficient for analysing patient records. We must also examine the sequence of the events and the time elapsed between them. Only once our analyses are utilising this extra information will we be able to maximise our impact when performing research on routinely-collected data.

Data science is a broad term for any theory or technique for analysing data. It encompasses methods from distributed computing, machine learning, computer science, statistics, mathematics, predictive modelling and data visualisation. However, most analyses performed under the heading of data science when applied to healthcare data do not fully make use of the available temporal information. A typical regression analysis might identify a cohort of patients and determine if any of a set of covariates were significant in predicting a certain outcome. While many of these techniques can provide useful analyses when applied to healthcare data, it is a missed opportunity if the information relating to event sequence and timings is neglected.

Wil van der Aalst [91], describes process mining as the bridge connecting traditional data science methods with the field of process science. He uses the term process science to describe “the broader discipline that combines knowledge from information technology and knowledge from management sciences to improve and run operational processes”. Process science includes disciplines such as operational research, business process management, stochastic analyses and optimization techniques.
Process mining comprises a set of tools and methods for reconstructing processes from event logs (see Figure 8). The underlying processes may be unknown, in which case this is process discovery. With known processes, process mining is used for conformance checking to assess how well the observed processes correspond to the expected process. Finally, process mining can be used to recommend process improvements.

Process mining has been applied across multiple fields and domains. Examples include: the analysis of an expense approval system within HP [92]; the analysis of the change management software process and a part review process at Mercedes Benz [93]; the analysis of the daily workflow patterns of a software team, and their response to support requests [94].

Figure 8 - Example care pathway showing the difference between the ideal process and the actual process for a fictitious surgical procedure.
5.2 Sequence Analysis

I wanted to explore novel ways of analysing a patient’s care pathway, or sequence of clinical events, to compare the outcomes of patients who followed best practice guidelines with those who diverged. This is similar to the conformance checking within process mining described above.

There are several ways that a patient’s care pathway might diverge from a standard: events that should occur may not occur; events that should not occur do occur; and events occur in an incorrect order. String matching metrics provide a way to measure the difference between two strings of characters by counting the number of operations required to convert one into the other. The permitted operations are: inserting a character; deleting a character; replacing one character for another; and swapping two adjacent characters. By mapping clinical events to characters, the sequence of events becomes a string and the distance to the expected sequence can be calculated using a variety of string metrics.

We explored several string matching methods using data from secondary care for a care pathway for the initial management of stroke. The methods used were: longest common subsequence (LCS); the simple edit or Levenshtein distance (LEV) which assigns weights to the operations of insert, delete and replace; and the Damerau-Levenshtein distance (DAM) which extends the Levenshtein distance by permitting adjacent character switching. I also added the concept of distance normalization for each of the methods. This was because the standard distance metrics would consider “cat” and “dog” to have a distance of 3 because 3 operations are required for conversion (c→d, a→o, t→g), but would also classify “cat on a mat” and “dog on a mat” as having a distance of 3 when the latter example is intuitively a closer match. By weighting the distance, by dividing it by the combined length of the two strings, we arrive at a distance metric that classifies the latter example as having a smaller distance between the strings.

For each patients’ record, we found the smallest distance for each string metric when compared against all possible pathways through the stroke model. We assessed whether the metric (a) matched the patient with their actual route through the pathway by giving it the shortest distance, and (b) whether the shortest distance was a unique
value rather than simultaneously matching multiple routes. We found that the LCS and DAM metrics correctly matched over 99% of patient routes, and that the normalized variants were best at generating unique results with normalized LCS providing a unique result for every patient route.

This worked well for the secondary care dataset, and my follow-up project was to explore how this would work when applied to primary care. However, this work preceded the work on meaningful events described in the previous chapter and there were barriers. Medications posed the largest problem as, without conversion to start and stop events, long-term prescriptions where the same medication was prescribed every few weeks dominated the care pathway strings. Two patients on the same pathway comprising a diagnosis followed by a medication should have a similar distance metric, but if one were on the medication for longer, then the difference would be large.

Another limitation was that we were not making use of the temporal information also available in EHR data. A patient who receives a diagnosis one year after a test may have diverged from the pathway more than the patient who receives their diagnosis within one day, but with the string matching method, this difference is lost. There are various ways to avoid this such as creating meaningful events that incorporate time: “patient is diagnosed within 1 week of test”, “patient is diagnosed between 1 week and 1 month after test”, and “patient is diagnosed more than 1 month after test”. This allows methods such as string matching to analyse sequences while incorporating temporal information. However, the obvious next move was to engage with process mining and utilise the methods there that use both sequence and timings.

5.3 Process mining

Before my work on process mining, there were already three literature reviews relating to process mining of healthcare data. Rojas at al. [95] reviewed 74 papers to explore: process mining techniques, tools and methodologies; implementation and analysis strategies; geographical region; and medical domain. Erdogan et al. [96] reviewed 50 papers to explore: the type of analysis; the frequency of use of various process mining algorithms; and the modelling notation. Ghasemi et al. [97] reviewed 168 papers
related to process mining in healthcare and reported on the growing trend for publications in this field. They also noted that the increasing proportion of process mining papers dedicated to healthcare.

Based on these reviews, and the papers reviewed in them, I suspected that most process mining work in healthcare was from secondary and tertiary care environments and not from primary care. This resonated with my previous experience attempting sequence analysis in primary care data, and I suspected that there were aspects of primary care data that presented challenges not present in other data.

I approached Eric Rojas to work with him to update his literature review, but this time with an extra focus on how the process mining related to primary care. This is publication 5 in this thesis. We found almost no studies that had used primary care data – see Table 2. Out of 143 papers that referred to a dataset, only 7 either had reference to primary care data or described their data in such a way that we could not rule out the possibility that it contained primary care data.

Table 2 – Out of 143 papers identified by our literature review, the number and proportion of papers performing process mining in each healthcare setting

<table>
<thead>
<tr>
<th>Healthcare setting</th>
<th>n (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hospital (secondary or tertiary care – but no primary care)</td>
<td>130 (91%)</td>
</tr>
<tr>
<td>Multiple domains including primary care</td>
<td>7 (5%)</td>
</tr>
<tr>
<td>Dentistry</td>
<td>4 (3%)</td>
</tr>
<tr>
<td>Nursing home</td>
<td>1 (1%)</td>
</tr>
<tr>
<td>Public health</td>
<td>1 (1%)</td>
</tr>
</tbody>
</table>

There are several reasons for this lack of research. In secondary and tertiary care environments there are obvious start and endpoints for a process, when a patient is admitted and when they are discharged. This is not true for primary care where much treatment, especially for chronic conditions, is open-ended. Secondary care also has tight temporal and physical boundaries with all patient interactions occurring within a short timeframe in a single geographic location. Interactions with primary care on the other hand are infrequent, sporadic and relate to the patient’s health when away from the point of care.
I attempted process mining using primary care data to identify the barriers and provide methods for other researchers to do the same. I focussed on pathways relating to medication safety. This was in part because of my work on the SMASH dashboard discussed in the next chapter. However, it was also because prescribing in UK primary care is almost entirely electronic, which means the data are very reliable and complete. I explored three processes around the prescription of non-steroidal anti-inflammatory drugs (NSAIDs) and antiplatelets to patient groups at a high risk of gastrointestinal bleeds (GiBs). These patients should receive gastro-protective medication (GPM) to mitigate the risk of GiBs.

My work on meaningful events allowed us to extract from each patient’s record the events we considered of interest to the process. This included when all the medications of interest were started and stopped. We also split GiB codes into two events: first bleed event, and subsequent bleed events. This allowed us to use the first bleed event to define a cohort of at-risk patients, and then use the subsequent bleed event as a negative outcome.

The attempt to use process mining in primary care was successful, and we discovered several things about the processes. GPM was far more likely to be co-prescribed with antiplatelets than with NSAIDs. NSAIDs were typically short-term prescriptions while antiplatelets were longer term. These two bits of information suggest that clinicians prescribing short term NSAIDs for acute pain relief consider the risk of GiB to be sufficiently low to not co-prescribe GPM, whereas patients on longer-term antiplatelets are perceived to be at a higher risk and therefore receive the GPM. GPM is not without side effects, so there is a cost associated with the prescribing.

5.4 Existing literature

Our review of 143 papers found seven publications that potentially used primary care data. Of the three that definitely included primary care data, two described themselves as preliminary analyses and only presented limited results [98,99]. The third [100] used a dataset combining healthcare administrative data with GP data for a cohort of patients with T2DM. Although this paper applied process mining to some primary care data, the results presented were simply two-stage processes documenting whether the
patient’s glycaemic control had increased, decreased or stayed the same. It is unclear whether process mining was useful in this case. However, the work they undertook to translate the HbA1c values into meaningful events such as whether a person’s glycaemia was high or low, and to consider consecutive values to classify events such as “High – increasing”, is the challenge and solution that we have been addressing in the papers presented here.

Litchfield et al. [101] produced a study protocol for exploring the use of process mining in UK primary care to study the processes of care around T2DM and hypertension. The authors acknowledge in their protocol one issue we have also highlighted; for chronic conditions, the “long-running processes have the potential to introduce complex cyclical models as similar sequences of events are repeated with variation as the disease progresses”.

The work by Mannhardt et al. [90] where the underlying event log is translated into an abstraction log of activities is relevant here. It was discussed in the previous chapter as it described a way of creating activities (meaningful events), but is reiterated here as the motivation behind their translation was to facilitate process mining. They applied their work to secondary care in the same way that we have done to primary care.

Similarly, Dagliati et al. [100] used an algorithm which grouped individual events under an “event type”. In their case, this was just “admission” and “short procedure unit”, but the idea is the same.

Finally, Lu et al. [102] consider the scenario where the same event label can refer to different events. One example they give is of a care process containing blood tests, consultations and x-rays occurring in multiple branches. A simplified approach using existing discovery algorithms will consider these separate events as the same thing because they have the same label, and produce a process map that does not correspond to reality. Their approach of pre-processing the event log to refine the event labels avoids this problem.
5.5 Impact of Publications

“Using String Metrics to Identify Patient Journeys through Care Pathways” was presented at the annual American Medical Informatics Association (AMIA) symposium. AMIA is the leading health informatics conference with each submission reviewed by three selected expert referees, one member of the SPC, and a member of the SPC leadership team. Reviewer comments included

- “Overall, the topic they are exploring is interesting and they present a useful approach.”

- “This is a very interesting paper presenting an innovative application of well-known computational techniques... The paper is well-written and the claims are supported by sound evaluation methods.”

- “the results for the stroke pathway examined in this study are impressive and promising”

“Process mining in primary care: A literature review” has been cited ten times. It has a ResearchGate Research Index of 10.4, which ranks it higher than 93% of publications from 2018. It was presented at MIE 2018 and received the following reviewer comments:

- “the article is well written and cleanly laid out. It should form a good reference to anyone engaging in process mining within healthcare.”

- “A very well written paper”

The impact of Publication 6 was described in the previous chapter.
6 Actionable information via dashboards

This chapter draws from publication 7 in the list of publications:

- SMASH! The Salford medication safety dashboard.

6.1 Background

An adverse drug reaction (ADR) is an injury that occurs when taking a medication. A recent report by Elliot et al. [103] estimated that avoidable ADRs caused 712 deaths and cost the NHS £100M per annum. The bulk of this is from primary care (£84M, 627 deaths) as this is where most prescribing occurs.

Hazardous prescribing is the act of giving a patient a medication that greatly increases their risk of an ADR. This might be an asthma exacerbation requiring hospitalisation for an asthmatic patient prescribed a non-cardioselective beta-blocker, or a gastrointestinal bleed in a patient prescribed anti-coagulants. Inadequate monitoring is the act of giving a patient a medication that requires regular blood tests to ensure a biomarker is within a normal range, but the monitoring is not performed, increasing the risk of an ADR. An example is methotrexate, which can cause blood dyscrasias and liver cirrhosis, so patients should receive a full blood count and a liver function test every 2-3 months [104].

Lists of indicators for hazardous prescribing and inadequate monitoring have been produced [105,106]. In a national study of 5 million patients, over 5% of eligible patients were affected by at least one prescribing indicator, and 12% of patients on relevant medication were not receiving adequate monitoring [107]. A study that examined the same indicators, but within the region of Salford, UK, found similar results [108].

The pharmacist-led information technology intervention for medication errors (PINCER) is an intervention aimed at reducing hazardous prescribing and inadequate monitoring. An RCT was performed that aimed to discover the importance of pharmacists in reducing hazardous prescribing [109,110]. All practices received feedback via lists of at-risk patients. The trial showed a large reduction at 6 months in
at-risk patients in practices that received their feedback via a pharmacist. Patients were less likely: to receive an NSAID if they had a history of peptic ulcer (OR 0.58 CI 0.38–0.89); to receive a beta-blocker if they had asthma (0.73, 0.58–0.91); or to receive an angiotensin-converting enzyme inhibitor or loop diuretic without monitoring (0.51, 0.34–0.78) [109].

The PINCER trial was a success in highlighting the importance of the role of the pharmacist in reducing the numbers of at-risk patients, and the intervention is also likely (59% chance) cost-effective [111]. However, the effect was not sustained at 12-month follow-up [109]. This is possibly because the feedback from the PINCER intervention provides snapshots of at-risk patients every six months from data extracted from the EHR. The effectiveness of feedback is improved when provided regularly and more than once [112]. I wanted to improve PINCER by increasing the frequency of the feedback, and improving its usability, to maintain the reduction of risk following the pharmacist’s initial intervention. I aimed to create a system that was updated daily, built upon the PINCER intervention, to provide information about at-risk patients. I planned to test the system in Salford, UK, as our earlier research had shown the prevalence of potentially hazardous prescribing there was 5.45%, and 7.65% for inadequate monitoring [108].

6.2 SMASH

Publication 7 describes the development and roll-out of a complex pharmacist-led dashboard intervention called SMASH (Salford Medication Safety Dashboard). Starting with a proposal from Darren Ashcroft (professor of pharmacoepidemiology) and Anthony Avery (the principal investigator of PINCER) I designed and developed SMASH. It comprises a server infrastructure that receives a feed of patient records from GP practices, processes the records, and applies the PINCER queries to generate lists of patients at risk of serious adverse outcomes. A secure web application presents this information to GPs and pharmacists.

I built the web application using an iterative approach, with prototype designs presented for feedback to a group containing six GPs, seven clinical commissioning
group (CCG) pharmacists, and one member of the public. Three iterations of this process led to the final design.

We hosted the server backend in the secure data centre at Salford Royal Foundation Trust (SRFT). The Salford Integrated Record (SIR) data warehouse, from which we received the feed of patient data, was also located there. The server infrastructure is a generic platform for receiving patient data and hosting patient-level e-A&F tools. The reuse of the infrastructure for a further two software tools, PATCHS and PINGR [35,113,114], confirms its utility.

![SMASH - The Medication Safety Dashboard](image)

**Figure 9 - Screenshot from the SMASH web application**

SMASH relies on the tools and methods described in the previous chapters. Code sets constructed in a transparent and reproducible manner allow the system to identify the conditions and drugs of interest accurately. Meaningful events for the starting and stopping of medications allow the system to determine when a drug is or is not in use by the patient, and therefore can precisely detect the at-risk patients. Event sequences, although not the primary focus of SMASH, proved useful for identifying different patterns of behaviour in the way patients were removed from risk. We can build this insight into future versions of the tool to increase its utility further.
To test the SMASH intervention, we recruited general practices from Salford. Practices were assigned a trained pharmacist, and the intervention started when the pharmacist first visited the practice. The pharmacist demonstrated SMASH to interested practice users, explained the importance of the indicators, and for an initial period, helped the practice to reduce the numbers of patients at risk. Following this exercise, the pharmacist would focus on another practice. We monitored practices for 12 months from this initial visit, during which time we recorded interactions with the dashboard together with aggregate data of the numbers of patients at risk for each indicator.

During the intervention, we undertook several additional studies. Jeffries et al. [115] interviewed a large sample of SMASH users including pharmacists, GPs, managers and nurses. The work highlighted the importance of the combination of information technology, the pharmacists, and the working environment. We concluded that SMASH had enabled a learning health system within the practices and was greatly appreciated, especially by the pharmacists whose quotes included:

“...the main benefit is that it’s just how quick and easy it is to access these patients.”

“...it’s just quick and easy... You can turn up at a surgery, log on the dashboard, and within an hour you could have made several safety interventions... you could have made quite an impact.”

Two studies were undertaken around usability and made use of eye-tracking equipment [116,117]. The first discovered a relationship between the movement and click events of the mouse, with the parts of the screen that the user was focusing on. This is important as it is infeasible to perform eye-tracking studies outside of the lab, but it is possible to track user interaction via mouse movements. The second study
showed a way to distinguish between different users of the system based on their *dwell time* (time between mouse clicks) and *exploration* (mouse movement). This suggests that it would be possible to develop systems that are responsive to the type of user to improve engagement further.

The final publication [118] looked at the impact of SMASH and found that it exceeded the initial PINCER trial by showing that the reduction in at-risk patients was sustained at the 12-month follow-up. The variation between practices also saw a marked decline.

### 6.3 Impact of Publications

“SMASH! The Salford medication safety dashboard” has been cited ten times since publication. The application itself has led to several other publications of which I am a co-author [115–120]. It has a ResearchGate Research Index of 8 ranking it higher than 75% of all items on the site, and higher than 90% of items published in 2018.

We see the true impact of SMASH in the sustained reduction of patient harm across Salford, where the number of patients exposed to potentially hazardous prescribing reduced by 41% and inadequate blood-test monitoring has reduced by 24% between 2016 and 2019 [118]. This impressive result has led to the decision to deploy SMASH across the whole of Greater Manchester. We project that this will lead to at least 11,000 fewer patients being at risk of a serious adverse outcome with a total cost saving of around £1M [109]. It should be noted that the cost associated with the time spent by the pharmacist is not yet taken into account. A full health economic evaluation is underway which will determine the true net cost saving.
7 Discussion

7.1 Main Findings

The work presented in this thesis encompasses several of the key steps required to make routinely-collected healthcare data ready for research and provide actionable information to clinicians. If care is not taken to understand and manage the data, then it has serious implications for the methods reproducibility of studies that make use of this data. Only by using appropriate tools and methodologies, developed in accordance with good software engineering principles, is it possible to create research that is open, transparent, and reproducible.

Clinical code sets underpin most work with routinely-collected healthcare data, as they are the first building blocks required for interaction with datasets. Errors here can be costly in subsequent analyses. Necessary care must be taken to ensure errors are minimized. I reviewed the literature around clinical code set engineering and developed the ‘term set’ methodology to improve this aspect of research.

Meaningful events are also a key step in translating the underlying patient record into a form that is more amenable to analysis. This is a step that most researchers will undoubtedly take, but may not report fully on what they are doing. By giving this a name, I have helped researchers to identify what they are doing and enable them to discuss it more widely.

Process mining is an emerging field of data science, which is perhaps too young to be fully utilised. The work that I have done in this area has shown that while it is a useful tool for generating hypotheses about data and for getting a quick overview of temporal processes, there is a lack of methods for quantifying divergence from care pathways. Too often, the answer is to show the results to a clinician to see what they think. Clinician involvement and domain knowledge are necessary, but when biases and arbitrary decisions are reached based on the opinions of a single person, rather than being more data-driven, there are bound to be problems. Our string matching work attempted to address this by producing actual numeric values via distance metrics, but
far more work is required before this discipline reaches full maturity and its benefits truly realised.

7.2 Software engineering

In this section, I will assess my work against the list of software engineering principles and best practices discussed in Section 1.5 (Figure 2).

A key principle of software engineering is modularization. Systems should be broken down into small independent units, or modules, each of which is responsible for one area of functionality. Breaking software down into modules is beneficial because it allows software development to be compartmentalized. A change to one part of a system likely requires only changes to, and knowledge of, a single module. It is helpful for team working as the need for a single person knowing the entire codebase is reduced or even eliminated. Having well-defined interfaces allows modules to be swapped out and replaced with a better alternative. Groups can then specialize in one particular area while reusing the work of others that is the best in their field.

For my application of process mining to healthcare data, it would have been possible to create a single package of code that extracted, translated, processed and analysed the raw EHR data. However, it would then be almost impossible for another researcher, or indeed myself, to use any part of the system in isolation. The benefits of well-constructed clinical code sets are far-reaching. Similarly, the utility of the ability to extract meaningful events from an EHR is not limited to process mining. Separating these aspects of the functionality into separate modules allows each one to be used independently. The final code comprised three modules: clinical code set creation; meaningful event extraction; and process mining. This demonstrates my adherence to item 1 in the list of software best practices (Figure 2). This modularization also greatly assisted the development of the web applications SMASH and PINGR where the modules for clinical code set development and medication event extraction could be reused again with little additional impact on time or cost.

All the software developed and reported on in this thesis was developed in line with current best practices. Following the openness of science, all software discussed here is publically available as open-source code (item 7 of the best practice list - Figure 2).
This allows other researchers to scrutinize the work in a way that is simply not possible when publication is limited to the traditional medium of the journal. It also allows other researchers to help develop the software. Rather than disparate individuals working on projects in isolation leading to a large duplication of work, by consolidating projects into open source repositories you can realise a significant scale-up of output with multiple researchers working to a common goal.

Working with a team of developers is preferable to working in isolation (item 8 of the best practice list - Figure 2). However, this has not always been possible, with most of the work reported here developed by just myself. However, my experience of software engineering has ensured that all the projects are structured to support team working.

All my work is version controlled with a strict versioning strategy (Figure 2 – item 2). This ensures that code is never lost and mistakes can be rolled back to a recent stable version of the software. It also ensures that multiple developers can work on elements of the software in isolation before merging their changes into the main branch following a code review. Issue tracking is used to keep a log of bugs that need to be fixed and enhancements that need to be added (Figure 2 - item 4). This is publically available so anyone using the software can add their own items. Each project has a well-defined suite of tests (Figure 2 - item 5), and a build process to ensure the source code is consistently compiled in the same way each time, to avoid the introduction of bugs (Figure 2 - item 3). Finally, and perhaps most importantly, each piece of software is well documented (Figure 2 - item 6), and the documentation is publically available for inspection. This allows others to reuse the software without first working out what it does and how to use it.

7.2.1 The Research Software Engineering Movement

Almost all scientific research depends on software. This has led to many software engineers working in academia. However, the importance of the software, and the engineers themselves, is rarely fully realised by the academic community [121]. Around the time when the first publication for this thesis was being drafted (2013), a movement was growing in the UK. A small group of software engineers working in academic institutions realised that they faced similar challenges. Universities are structured around academics, leaving software engineers as second-class citizens and
their importance diminished. Their role is often neglected, with a survey finding that despite 88% of software engineers in academia contributing to publications, 30% were not named as authors, and 24% were not even acknowledged [122]. They also rarely have a recognised career path within academia because of their non-academic role. It has been suggested that software should be seen as a “first-class experimental scientific instrument” to enable better research, but progress to this goal is slow [123].

A survey in 2014 found that out of 400 job adverts for academic roles that required software development, there were 200 different job titles. It was clear, that to establish a movement focussing on software engineers within academia, a name was needed. This was the birth of the Research Software Engineer (RSE) [124], a title which describes my current role. The RSE movement began in the UK but it has now spread internationally [125]. The Society of Research Software Engineering is now a registered Charitable Incorporated Organisation in the UK with the following goal [126]:

“The Society of Research Software Engineering was founded on the belief that a world which relies on software must recognise the people who develop it.”

Whether in academia or industry, software needs to be sustainable [127]. It should be built in a way that ensures its longevity. This can be challenging in a research environment when grant money runs out, or software developers change jobs. Almost half of RSEs (46%) surveyed work on software projects with a bus factor of one; only one person would need to be “hit by a bus” for the project to halt because of a lack of people with sufficient knowledge for it to continue [122]. In parallel with the birth of the RSE, several academic organisations have been established with the explicit goal of improving best practices around software sustainability. These include the Software Sustainability Institute, the Better Scientific Software, and Software Carpentry [128].

When considering the role of the RSE within academia, it is also useful to consider the difference between scientific coding, which is the use of software by a scientist, and the practice of software engineering. Scientists are often clever people and
frequently learn how to code for their research. However, the exploratory nature of science is at odds with the rigour of software engineering. Both are important. Researchers can undertake a fast, iterative approach to coding to make discoveries, while good software engineering principles can be used to build robust software for areas of science that are already known [129].

7.3 Implications and Recommendations

There is a growing consensus within the field of health informatics that the gap between routinely-collected data and research is one that needs attention. A few years ago the best reporting guideline for observational studies was the Strengthening the Reporting of Observational Studies in Epidemiology (STROBE) statement [130]. However, in 2015 this was acknowledged as insufficient for observational studies using routinely-collected EHR data and the RECORD checklist was created [33]. Checklists such as these, aim to achieve standardization in the reporting of such work. However, a recent survey found that out of researchers who performed observational research, nearly 40% had either never heard of STROBE, or had heard of it but never used it [131]. It is not unreasonable to assume that the knowledge and adoption of RECORD will be worse, as the checklist is only 5 years old versus 12 years for STROBE (in 2020). It is good that such checklists exist, but unless journals insist on their completion, it is hard to imagine that researchers will follow them routinely.

Recommendation #1: Journals should:

1. insist on the submission of a completed RECORD statement for studies using routinely-collected data;

2. ensure that the RECORD statement is peer-reviewed.

There are many barriers to achieving reproducible research. If I take a publication on observational research, even one that has followed the RECORD statement, I will still likely face problems if I attempt to reproduce the methods and the findings. Unless the researchers have shared all of their ETL code, I may still need to make several assumptions about their methods. Item 22 of the RECORD checklist states that “Authors should provide information on how to access any supplemental information such as the study protocol, raw data, or programming code” [33]. However, it needs
to go further and state that all code should be made available with clear instructions in the completed RECORD checklist on how to access it. This sharing is not limited to the analysis code, but should also include the code required to extract, clean and translate the underlying data.

**Recommendation #2: Publishing of code**

1. Researchers should aim to make their data cleaning and analysis code publically available.
2. The RECORD statement should be extended to ensure that all data cleaning and analysis code is made publically available.

Even if I do have access to the code, the sensitive nature of healthcare data means that I may not have access to the same data source. Without access to data in the same format as dictated by the inputs required by the code, it will be hard to execute the code.

**Recommendation #3: When data is not reasonably available to others, then researchers should ensure that:**

1. the underlying database schema is made publically available;
2. a test set of dummy data is provided to ensure other researchers can execute quickly the data cleaning and analysis code

If the researchers have shared their code, but without proper documentation, it may be hard, if not impossible to run. Therefore, researchers should ensure that their code is well documented. This should include comments in the code itself, explaining what each section does, and higher-level documentation which explains how to set up and execute the software.

**Recommendation #4: Documentation:**

1. Researchers should aim to provide high-quality documentation alongside their code.
2. Documentation should be peer-reviewed before publication to ensure it is understandable and of sufficient quality.
As third-party libraries and programming languages change over time, code becomes stale and eventually stops working unless it is continually updated. Containerisation software, such as Docker, will increasingly become used in this area, as it is a mechanism to encapsulate a software project and all of its dependencies into a single package which can then be run on any environment [132]. Making software open source is another helpful step to reduce the chance that it will become obsolete or cease to be used. However, for true sustainability, each open source project requires a community of developers, working in different institutions and across different grants, to increase the “bus factor”. Unfortunately, this is not simple and requires a lot of time and money. Research funders need to realize this to ensure adequate money can be provided for research grants that have software as an output, to ensure the necessary time is spent to increase the sustainability of the software beyond the lifetime of the project.

**Recommendation #5: Sustainability**

1. Researchers should consider making their code available via a container, such as Docker, to ensure its longevity.

2. Researchers should consider making their code open source

3. Research funders should make extra money available for research with software outputs.

Successful software engineering is critical to successful research, and both researchers and software engineers need to improve in the areas highlighted above. However, individuals in academic institutions alone cannot solve these problems. As previously mentioned, journals and research funders have a role to play, but so do Universities, who need to do more to acknowledge the existence and importance of the RSEs so that their role is not secondary to that of researchers.

**Recommendation #6: Ecosystem**

1. Research software should be developed in accordance with the principles and best practices of software engineering.
2. Projects with software outputs should ensure that a Research Software Engineer is part of the team.

3. Universities need to acknowledge the existence and importance of RSEs by providing market-competitive salaries and well-defined career paths.

The recommendations provided here are not straightforward, and almost always would require extra effort, particularly by the researcher. This will be challenging unless the benefits of such an approach can be properly articulated and quantified. It is also hard to imagine research funders making more money available for software engineering unless an economic evaluation had shown that it would achieve a net benefit. While the lack of methods and findings reproducibility is well documented in the literature, what is unknown, is the extent to which observational research with routinely-collected data is reproducible, and the effort involved in its reproduction. There would seem to be a complete absence of empirical reproducibility research literature. Therefore, research should be undertaken where RSEs and data analysts attempt to reproduce a random sample of studies, then report on the extent to which the methods and the findings could be reproduced, together with the effort involved in terms of time, resources, and people. Such research is necessary to highlight the extent of the problem, and enable further research to focus on the economic impact and benefits that could be achieved if the duplication of effort due to a lack of reproducibility was avoided.

**Recommendation #7: Research**

1. Future research should attempt to assess the degree of reproducibility in existing published studies.

2. Future research should attempt to quantify the cost associated with a lack of reproducibility, and the benefits that could be achieved if improvements were made in this area.

Ideally, researchers would use tools that allowed them to adhere to the above recommendations, while simultaneously making their job easier. Persuading researchers to adopt tools that provide them with a direct and immediate benefit, is easier than asking them to follow processes that might benefit other researchers at a
future date. My GetSet tool for creating code sets aims to do just that. For most people, the tool allows them to create code sets in a way that is simpler than their existing processes while also producing an improved output.

For the full research lifecycle, it is worth considering the work into Research Objects [133,134] and eLabs [135]. Research Objects provide a mechanism so that the elements of research which are not in the final manuscript, such as the data and analytic code, can be packaged in a machine-readable way. This enables research to be shared in a way that facilitates methods reproducibility. ELabs go further by providing a platform to bring together researchers, methods and data. All work is conducted in the platform and therefore reproducibility can be as simple as clicking a button to rerun an analysis.

Research Objects and eLabs are challenging the traditional model of research, where the final output to be shared with the research community is the published paper. The methods and results of any analysis conducted on a computer are obfuscated by translating them into the written format of a manuscript that is human- but not machine-readable. The description of what was done is still necessary for our understanding and critique of the work, but if the primary output was machine-readable, rather than a traditional manuscript, then it would be a big step forward towards universal methods reproducibility.

We can, and will, solve these problems, but only by a combined effort across the research community.

7.4 Strengths and Limitations

Transforming routinely-collected healthcare data into research-ready data is a form of software engineering. The methods and software developed in the publications submitted for this thesis were developed in accordance with good software engineering principles. Therefore, the main strength of my approach in this thesis is that, as a Research Software Engineer, I have been able to apply the principles of software engineering to a part of research where it is typically lacking.

One limitation is that for most projects I have been working alone. Collaboration is a key part of good software engineering, with techniques such as “code review” and
“pair programming” becoming widespread in industry. Critically appraising the code of others, and in turn, being critically appraised leads to improved and faster learning, and code that is more reliable. This is not always possible in the research environment. However, I have endeavoured to put in place the same processes that would be required for collaborative working, and have made all software publically available and open source. This means that, should others wish to contribute to any projects, there are no barriers to this happening. Indeed, several of the projects have now had involvement from other people in the research community who have raised issues and contributed new features.

Another limitation is that I could not find a definitive list of software engineering principles or best practices, against which to assess my work. This led me to construct my own list, which could be seen as a weakness of the thesis. However, by modifying the list of best practices for scientific computing by Wilson et al. [37], I have provided a foundation that increases the credibility of the list. Also, I have not simply selected items for which I could provide evidence, as demonstrated by my lack of collaboration; an important best practice of software engineering absent in my work.
8 Conclusion

The vast amount of routinely collected healthcare data that is now available for research presents us with fantastic opportunities. Intelligent clinical decision support and electronic audit and feedback can assist clinicians in implementing best practice and reduce the risks of harm done to patients. Retrospective observational studies, which are both cheaper and easier to perform than randomised control trials, can show the effect of interventions in a real-world setting while avoiding the biases inherent in RCTs where large populations such as children, pregnant women, the frail and the elderly are routinely excluded.

However, caution is essential as all too frequently publications omit the steps they have taken to craft the raw data into the final dataset used for their analysis. Where methods and tools exist, they are not always developed in strict accordance with the principles and best practices of software engineering. The work presented in this thesis goes some way towards bridging the gap as I have applied my expertise in software engineering to the problem of working with routinely collected healthcare data. Only with the appropriate methods and tooling, developed with the rigour of software engineering, can we remove the barriers to truly reproducible research and rapidly increase the speed of discovery in order to improve and extend the lives of people.
References


[49] J.M. Hogan, R. Thomas, Developing the Software Engineering Team, in:


[54] SNOMED International, Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT), (n.d.).


[116] A. Yera, J. Muguerza, O. Arbelaitz, I. Perona, R. Williams, N. Peek, R. Keers,


[127] C.C. Venters, R. Capilla, S. Betz, B. Penzenstadler, T. Crick, S. Crouch, E.Y. Nakagawa, C. Becker, C. Carrillo, Software sustainability: Research and


Appendix A – Published Work

Publication 1: Clinical code set engineering for reusing EHR data for research: A review.
Methodological Review

Clinical code set engineering for reusing EHR data for research: A review

Richard Williams, Evangelos Kontopantelis, Iain Buchan, Niels Peek

Objective: To review methodological literature on the management of sets of clinical codes used in research on clinical databases and to provide a list of best practice recommendations for future studies and software tools.

Methods: We performed an exhaustive search for methodological papers about clinical code set engineering for re-using EHR data in research. This was supplemented with papers identified by snowball sampling. In addition, a list of e-phenotyping systems was constructed by merging references from several systematic reviews on this topic, and the processes adopted by those systems for code set management was reviewed.

Results: Thirty methodological papers were reviewed. Common approaches included: creating an initial list of synonyms for the condition of interest (n = 20); making use of the hierarchical nature of coding terminologies during searching (n = 23); reviewing sets with clinician input (n = 20); and reusing and updating an existing code set (n = 20). Several open source software tools (n = 3) were discovered.

Discussion: There is a need for software tools that enable users to easily and quickly create, revise, extend, review and share code sets and we provide a list of recommendations for their design and implementation.

Conclusion: Research re-using EHR data could be improved through the further development, more widespread use and routine reporting of the methods by which clinical codes were selected.

© 2017 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).
1. Introduction

The rapid adoption of electronic health records (EHRs) is creating unprecedented opportunities for studying clinical medicine, health services, and population health [1–3]. Adoption of EHRs has more than doubled in the US over the last decade and is expected approach 100% in 2020 [4]. The rapid adoption of EHRs is creating unprecedented opportunities for studying clinical medicine, health services, and population health [1–3]. Adoption of EHRs has more than doubled in the US over the last decade and is expected approach 100% in 2020 [4]. UK primary care providers switched to using EHRs in the 1990s, and this has made large EHR databases available for research [5,6]. The re-use of EHR data for research provides large datasets with long follow-up times against low costs [7–9]. However there are considerable methodological challenges because raw EHR data are not “research ready”: they need to be transformed before they can be meaningfully used for research [10]. Such transformations typically involved the automatic extraction of clinically meaningful features from the raw data. Other problems are the large variation in coding practice between clinicians [11] and the effects of coding incentivisation schemes which aim to improve coding behaviour [12].

EHRs typically contain structured data (clinical codes for problems; diagnoses; treatment; and management) and unstructured data (free text; images). Most studies that re-use EHR data for research focus on the structured part of the EHR, thus using clinical codes to extract meaningful information from the raw EHR data. Clinical code set construction is the process of assembling a set of clinical codes that represent a single clinical concept such as a diagnosis, a procedure, an observation or a medication. Once constructed the set is then used to query and extract data from an EHR database, for use in further analysis. These sets of codes, variously referred to as “code lists”, “clinical code lists”, “code sets” and “value sets”, are one of the main building blocks for creating the database queries and would typically be used to: identify a cohort of patients for use in an observational or retrospective study; identify the covariates of interest, confounders and endpoints for the same observational study; or identify potentially at risk patients when used as part of a clinical quality measure [13]. The size and complexity of the final set will be determined by a combination of the particular code terminology and the concept of interest. An evaluation of 1054 code sets created for clinical quality measures found the size varied from sets containing a single code, to a code set for trauma which included 20,560 codes [14]. The construction and validation of such lists is a non-trivial matter.

The creation of sets of clinical codes for querying routine healthcare databases is an important part of using such data for research. It is frequently an early step in research, and arguably a hazardous one, as errors introduced here of missing or wrongly specified codes could result in selection biases that propagate throughout subsequent analyses having a major impact [15]. For instance, Muller and co-workers found that code set differences could induce nearly a sevenfold difference in estimates of the incidence of rheumatoid arthritis [16,17]. The re-use of EHR data for research is a relatively new field, and so far there has been a lack of awareness within the community that code set engineering is an issue that requires attention. This is reflected in the lack of methodological literature on the subject despite its importance [18,19]. Recently, the area has received more attention with Gulliford et al. calling for “greater transparency in the selection of lists of codes for different conditions, as well as sharing of code lists among researchers” [20]. However, several reviews have shown that code sets are rarely included in applied papers [19,21], let alone the process by which the code sets have been constructed. The construction process is arguably of greater significance than the set itself as the process could, and should, be scrutinised by reviewers, and can form the basis for other researchers to reuse their methods rather than taking a crude set on trust.

1.1. Objectives

This review has two main objectives:

1. to review and compare methods and tools for managing (constructing, validating, sharing, and reusing) sets of clinical codes reported in the literature; and
2. to develop recommendations for the management of clinical code sets and for the design and implementation of clinical code set management tools.

To the best of the authors’ knowledge, these objectives have not been examined in depth previously, therefore this review enriches the literature and can be used to inform future work in this area.

2. Background

2.1. Clinical terminologies

A clinical terminology, or complex dictionary, is a structured collection of codes and descriptions used to represent aspects of clinical practice and can include: diagnoses, procedures, medications, laboratory results and administrative data. Examples include the Systemized Nomenclature of Medicine – Clinical Terms (SNOMED CT) [22] and the International Statistical Classification of Diseases and Related Health Problems (ICD) [23]. United Kingdom (UK) specific codes include Read version 2 [24] for capturing...

Terminologies are typically represented as a taxonomy containing a hierarchy of codes or concepts. This can be done simply, as with Read version 2 or ICD-10-CM where code prefixes determine ancestry, so that in Read: “G... – Circulatory system diseases”, is a parent of “G3... – Ischaemic heart disease” which in turn is a parent of “G30... – Acute myocardial infarction”, while in ICD-10-CM: “Ix... – Diseases of the circulatory system”, is a parent of “I21 – ST elevation (STEMI) and non-ST elevation (NSTEMI) myocardial infarction”, which in turn is a parent of “I21.4 – Non-ST elevation (NSTEMI) myocardial infarction”. This method of hierarchy, while simple, is restrictive and prevents a concept from having multiple parents thereby resulting in duplicates of terms or inaccuracies. CTV3 is an example of a taxonomy where the hierarchy is stored independently of the codes which allows for multiple inheritance.

While the only relationship in a taxonomy is the parent/child or “is a” relationship, an ontology, such as SNOMED CT, allows arbitrary relationships to be defined. This not only allows for multiple inheritance but also relationships such as “has finding site” or “has causative agent” [22]. While sufficiently expressive to cover all clinicians’ coding requirements, the complexity of the ontology requires novel searching techniques [26]. There are several publicly available online services that allow users to search and navigate multiple terminologies such as several tools from the Unified Medical Language System (UMLS) [27] and the BioPortal from the National Centre for Biomedical Ontology [28].

2.2. The language of clinical code sets

A clinical code set in its most basic form is simply an unordered group of distinct clinical codes taken from a clinical terminology. However the term “clinical code set” is not a universally recognised term, and is variously referred to in the literature as a “clinical code list” [18,19,29], a “value set” [14,30–32], or simply a “code list” [23–35]. All the codes in a set would typically relate to a specific medical concept and these codes would be used by researchers to construct queries to execute against a database to extract patients or data for use in further analysis. The codes in the set are interpreted as a disjunction, meaning that a code set describing hypertension containing codes A, B and C would be used to search for patients with any of the three codes. For simplicity, we will refer to these sets of clinical codes as “code sets” throughout the remainder of this document.

2.3. Electronic phenotyping

Electronic phenotyping algorithms are designed to extract clinically meaningful features from raw, routinely collected EHR data [36]. Next to coded data they typically use data on prescribed medications, procedures, laboratory tests and administration codes; free text which is processed with natural language processing (NLP) techniques; and occasionally genomic data. There are many examples of large scale electronic phenotyping projects such as Electronic Medical Records and Genomics (eMERGE) [37], and platforms that facilitate electronic phenotyping such as Informatics for Integrating Biology and the Bedside (i2b2) [38] and the Observational Health Data Sciences and Informatics (OHDSI) program. Such phenotyping systems and supporting platforms are typically designed as a data warehouse or as a distributed/federated system.

The ability to accurately create clinical code sets in a transparent and reproducible manner is of paramount importance for the credibility of electronic phenotyping and so it is therefore necessary to include in this review any such reported methods.

Electronic phenotyping describes a broad methodology rather than a specific process and can be achieved in a multitude of ways which are often system specific. The ability to create, validate, reuse and share a code set is a core component of electronic phenotyping platforms, retrospective observational studies, database analyses and quality measures. It is therefore useful and important to study clinical code set management in isolation from the settings in which it is found.

3. Method

3.1. Scope

Literature that may be of use to this review can be thought of in two categories:

1. methodological – a paper that discusses or presents methods or tools for the construction or sharing of a code set;
2. applied – a paper that makes use of a code set for a piece of analysis.

These categories are by no means mutually exclusive; some applied papers will explain their methods or provide code sets, and some methodological papers will attempt to apply their methods as a form of validation [29].

This literature review aims to exhaustively search for papers with a substantial methodological component. Applied papers with little to no methodological component are considered out of scope of this review; this is discussed further in the limitations below.

3.2. Bibliographic search process

Search terms were chosen based on the assumption that any paper discussing methods for managing clinical code sets, would include a synonym for “code set” within their title or abstract.

One such synonym, “value set”, can more generally mean a set of possible values to choose from when answering a question, and frequently occurs in literature concerning EQ-5D® (a widely used instrument for measuring health-related quality of life [40]), which is not related to clinical terminologies. So, the term “EQ-5D” was excluded and the final search criteria were:

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Justifications for papers rejected after the review of the abstract or full text (n = 95).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reason for rejection</td>
<td>n</td>
</tr>
<tr>
<td>An applied paper, or one that uses code sets but with no discussion on the management (construction, reuse, validation, sharing) of code sets</td>
<td>27</td>
</tr>
<tr>
<td>A paper with a higher level focus on terminologies such as mappings between them, but no focus on code sets</td>
<td>26</td>
</tr>
<tr>
<td>A paper covered elsewhere – either a paper on a phenotyping system found during the methodological search but included in the phenotyping search, or for example two pieces of very similar work by the same group where the later work supersedes the earlier</td>
<td>13</td>
</tr>
<tr>
<td>A paper where the phrase “value set” or “code set” appears, but where the word “set” is used as a verb rather than a noun</td>
<td>8</td>
</tr>
<tr>
<td>A paper specifically discussing some aspect of the transition in the US from ICD9 to ICD10</td>
<td>6</td>
</tr>
<tr>
<td>A paper on coding practice amongst health care workers</td>
<td>4</td>
</tr>
<tr>
<td>A paper where the mention of code sets was very brief – just one or two sentences</td>
<td>2</td>
</tr>
<tr>
<td>A paper more than 20 years old considered to be out of date</td>
<td>2</td>
</tr>
<tr>
<td>A paper for which the abstract of full text was unavailable</td>
<td>2</td>
</tr>
<tr>
<td>A paper specifically discussing some aspect of the transition in the US from ICD9 to ICD10</td>
<td>2</td>
</tr>
<tr>
<td>A paper discussing code sets, but not clinical code sets</td>
<td>1</td>
</tr>
</tbody>
</table>
1. papers whose abstract or title contains any of the terms: “code list(s),” “code set(s),” “value set(s),” “list of codes,” “set of codes”; and
2. papers whose abstract or title does not contain the term “EQ-5D”.

An initial search on PubMed was performed in August 2016 by the lead author using the query: (“code list” [All Fields] OR “code lists” [All Fields] OR “code set” [All Fields] OR “code sets” [All Fields] OR “value set” [All Fields] OR “value sets” [All Fields]) NOT “EQ-5D” [All Fields]). An additional search was performed on OvidSP for the terms “list of codes” and “set of codes” as we were unable to perform this on PubMed due to a limitation whereby certain common words (the, a, of) are not indexed, and cannot be searched for even when within a quoted string of text. OvidSP is an alternative search interface for the MEDLINE database that allows quoted phrases to include common words and so was better suited for these search terms.

The list was supplemented with papers identified by searching citations of relevant material, via snowball sampling using PubMed and Google Scholar. Papers were excluded if they were not related to clinical code sets or not written in English.

In total, 507 papers were discovered from the initial searches with 502 screened after 5 duplicates were removed. 464 papers were rejected for lack of relevance (435 from the title and 29 from the abstract). The snowball sampling commenced from the 38 full papers read and in turn this identified 31 additional papers (initial list of 183 papers was reduced by removing 26 duplicates, 109 rejected from the title and 17 rejected from the abstract). Out of the 79 full papers reviewed, 49 were rejected as not relevant. Justification for the 95 papers rejected from the abstract or full text is given in Table 1. This review is based on the remaining 30 papers. The full process can be seen in Fig. 1.

In addition, it was deemed appropriate to review methodological papers on electronic phenotyping systems because they may contain details of the component responsible for code set management. We were aware of several systematic reviews of electronic phenotyping systems [41–44] and therefore constructed a list by collating all such systems mentioned in each review. A total of 36 electronic phenotyping systems were identified from the four review papers, however six were deemed unsuitable for the review because they described systems that didn’t use coded data. Papers or websites describing the architecture of the remaining 30 systems where then search for.

All papers reviewed are listed in Appendices.

3.3. Data extraction

The primary area of focus was any information relating to the management of code sets, either their construction, validation, sharing or reuse. This was anticipated to include: strategies for code set construction; descriptions of software tools and methods; platforms for sharing code sets; and discussions of relevant approaches. However the strategy for concept and data extraction from the reviewed papers was deliberately left open ended to allow anything of interest to be captured.

The lead author performed two reviews of each paper. During the first iteration each paper was tagged according to the presence
of existing concepts. New concepts discovered were added to the list of existing concepts and looked for in future papers. The second iteration ensured that concepts identified latterly, were discovered in earlier papers if present.

We also explicitly aimed to extract the code terminologies and the country of origin of the databases used by the papers.

4. Results

4.1. Methodological literature

Although differences existed between the methods described in the 30 methodological papers, common themes emerged (Table 1). A popular approach was to reuse an existing code set (n = 21) from: a previous study (n = 5); a national clinical quality management scheme such as the Quality Outcome Framework (QOF) in the UK and the Value Set Authority Centre (VSAC) in the US (n = 11); or both (n = 4). The reused set was almost always updated or extended (n = 20). Authors reported some specific strategies for searching for relevant codes: exploiting the hierarchical nature of coding terminologies (n = 23); preparing an initial list of synonyms to search for (n = 20); and employing an iterative approach after preliminary searches (n = 13). The putative sets were usually reviewed (n = 26), mostly with clinician input (n = 20), before definitive use.

To give an example of some of the steps described in Table 1 we will summarize the approach taken by Olier et al. [29] to construct a code set for the identification of patients with severe mental illness (SMI). The QOF business rules, which give lists of Read v2 codes to be used by clinicians for coding certain concepts, was used as a starting point, along with a list of synonyms provided by a panel of clinicians. The hierarchy of the Read code v2 dictionary was used to discover additional codes that corresponded to children of the codes identified from the original QOF list, and the list of synonyms was used to search the dictionary for additional codes. This led to a preliminary list of 506 read codes which were then independently reviewed by the panel of clinicians and reduced to a conservative list of 270 codes by consensus. A larger, more speculative list, was also prepared to allow sensitivity analysis to be performed. The paper indicates that additional iterations of the approach could be performed if extra synonyms were discovered, however in the case of SMI it was not necessary.

The sensitivity and specificity of codes sets, i.e., the degree to which the code set correctly identifies patients with a particular condition, was a frequently recurring topic (n = 19). Sometimes this was in terms of the codes selected during the search phase with one paper commenting that sets of codes were required as single codes may not be sufficiently sensitive or specific [45], while another aimed for high sensitivity, as the final set would be subject to a manual review [49]. Usually however the discussion was in the context of the patients identified via the generated code set. The different demands for sensitivity and specificity were discussed and how this would differ depending on the study [19], with a couple of papers noting that while quality measures might focus on high sensitivity to ensure that at risk patients were not missed, e-phenotyping or cohort selection procedures would be more focused on a high positive predictive value (PPV) [18,36,58].

There were frequent suggestions for a sensitivity analysis [20,29,54,65,66] where multiple sets of varying degrees of conservatism would be created and compared in the final analysis. Another paper noted the lack of validation studies performed on code sets which could be key to determining their sensitivity and specificity [48]. One final paper commented on a problem specific to routinely collected data whereby “the absence of a ... code for disease must be interpreted as an absence of the disease itself, so whereas positive predictive value tends to be high, sensitivity may be lower” [5].

Several papers urged caution due to the temporal and dynamic nature of code sets (n = 13). The underlying prevalence of a condition can change over time [65], or coding practice can change irrespective of the underlying prevalence [45,65], perhaps due to changes in the software used to record the codes. The understanding of a condition can change, and also schemes that incentivise good coding practice can alter clinician behaviour in the areas where funding is offered [20,65].

Seven papers described software to support the selection of code sets, of which three used a data-driven approach such as a random forest [58–60], and three are open source and publicly available and discussed below. A further two papers suggested features for such tools, that software should: be backwards compatible [36]; “support search by similarity within and across code systems”; provide “maintenance when code systems evolve”; allow “intentional definitions (e.g., “diabetes mellitus” and all its descendants in SNOMED CT), as well as extensional definitions (lists of codes)”; and respect the particular code formatting of a terminology [50].

There were frequent calls for openness and the sharing of code sets and code set management methods (n = 14) such as: “the need for transparency in the reporting of case definitions” [54], “a need for greater transparency in the selection of sets of codes for different conditions. . . as well as sharing of code sets among researchers” [20], the need for “an authoritative source of value sets” [50], the benefits of the “inclusion of . . . code sets and a description of the process used to develop code sets in publications” [64], and that “the publication of precise search terms and results, may allow transparency and replicability in the preparation of code-lists and facilitate their sharing” [46]. One paper suggested that code sets may not be shared either because the authors had not considered it an important aspect of their research, or that they may be protective of the time it had taken to construct the set and be unwilling to share as other groups may reuse the set without due credit [34]. Some papers gave actual suggestions or platforms for sharing (n = 7) and the value of creating metadata alongside a code set was highlighted as a way to assist with sharing by facilitating other researchers to discover and re-use a set [19,50].

The sharing of the final code sets is often neglected [19,64], or done in an ad-hoc fashion as an appendix or supplement to a paper or stored online on the author’s university web page. One paper observed that the large size of code sets made them unsuitable for inclusion in a paper and should be hosted online in a curated repository [64]. Occasionally this occurs with the code set being uploaded to a platform for sharing code sets such as Clinicalcodes.org [67] or as part of a phenotype definition in repositories such as PheKB [37], phenotypeportal.org [68]. However, these sites are not well used. The “phenotypeportal” link no longer works and forwards users to the PHEMA homepage, PHEKB only has 34 publicly listed phenotypes, and ClinicalCodes, although the most popular with 40 papers and 370 code sets, has minimal usage from outside the University of Manchester where it is hosted.

Most papers used a dataset from the UK (n = 18), with some from the US (n = 8), and the rest (n = 4) described methods without reference to a country or database. The terminologies used can be seen in Table 2. The high proportion of UK databases explains the high proportion of Read v2 codes Table 3.

4.2. Electronic phenotyping literature

The electronic phenotyping systems (n = 30) investigated were either designed as a data warehouse (n = 18) or as a distributed system (n = 12). Of the data warehouses 11 were for a single institution or system, with seven designed for multiple institutions and
All the distributed systems were, by definition, designed for a multi-site architecture. Just over half (n = 16) of the reviewed systems provided any detail about the code set construction process involved: a list of synonyms was created (n = 15) and then the act of constructing code sets is left in its entirety to the end user, such as with 12b2 [38] and eMERGE [37]. In the middle systems like PCORnet [73], require that all local terms are mapped to a standard vocabulary or terminology, with construction of code sets limited to the mapped vocabulary.

Where systems allowed the sharing of information, this was always done by sharing the entire phenotype definition, rather than the code set in isolation.

### 4.3. Existing software tools

Davé and Petersen [18] gave details of the Stata script that they used for constructing code sets. A researcher and a GP first identify a list of synonyms and word stems (e.g., diabet*) in order to match diabetes and diabetic) which are then searched against the descriptions in a dictionary. Code stems are also searched for, and as they are using Read codes with prefixes determining ancestry, this is the equivalent of a hierarchical search. They highlight that the Stata script used to create the code set is self-documenting and can be used by others to scrutinise the process by which the code set was created.

Olier et al. [29] presented an R package (Rpcssearch) and equivalent Stata command (pscssearch) for constructing a code set. The process is very similar to the one proposed by Davé and Petersen where code stubs and word stubs are used as the basis of an iterative search, however they go further by presenting fully documented and more flexible approaches since they allowed the use of Boolean operators in the search. They also suggest that the final code sets should be reviewed by an expert panel. They then shared their resulting code sets for various projects via clinicalcodes.org.

The Clinical Research using Linked Bespoke Studies and Electronic Health Records (CALIBER) [61] is an e-phenotyping system which has developed the R package CALIBERcodelets [74] to facilitate the construction of code sets. As with the previous two tools, users can search for code sets by synonym or code stub. Users can also combine search terms with Boolean operators, and the tool supports regular expressions which allow for more complex search queries. The final output is a csv file with the set of codes and some basic metadata such as the name and version of the code set.

Although not mentioned, the R script used to generate the set of codes for both Rpcssearch and CALIBERcodelets could be shared in addition to the code set itself to allow other researchers to “check the working” in the way advocated by Davé and Petersen.

Most tools used for code set construction in published e-phenotyping systems are not available for use outside of the host centre, and are often not described in sufficient detail to comment on their functionality. Code searching is usually an integral part of a wider system which is tightly coupled to the host database, rather than a module which can be used in isolation. The two exceptions are: ATLAS [75] developed by the OHDSI, and a tool developed as part of the Phenotype Execution Modelling Architecture (PHEMA) [76] project. ATLAS used a shopping cart analogy for constructing code sets and allows searching across an impressive array of more than 30 code terminologies. A simple text search returns a table of matches which can then be filtered by category such as: medication, procedure, and diagnosis. Terms returned can be grouped into “concept sets” and then exported in a variety of formats. It doesn't however support hierarchical searching, and no metadata is captured as to what was searched for. The PHEMA tool allows users to select a predefined code set from VSAC, or to perform a text search for ICD10 codes which are then displayed in a hierarchy and can be selected and exported.

---

**Table 2**

Number of occurrences of common themes from the reviewed papers.

<table>
<thead>
<tr>
<th>Item</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Papers reviewed</td>
<td>30</td>
</tr>
<tr>
<td>- Reuse existing code set</td>
<td>21</td>
</tr>
<tr>
<td>- from a previous study</td>
<td>5</td>
</tr>
<tr>
<td>- from a national clinical quality management scheme</td>
<td>11</td>
</tr>
<tr>
<td>[14,29,31,36,49–55]</td>
<td></td>
</tr>
<tr>
<td>- from both [19,20,36–58]</td>
<td>5</td>
</tr>
<tr>
<td>- Updates an existing set</td>
<td>20</td>
</tr>
<tr>
<td>[14,19,29,30,34,36,45,48,50–59]</td>
<td></td>
</tr>
<tr>
<td>- Exploiting hierarchical nature of coding terminologies</td>
<td>23</td>
</tr>
<tr>
<td>[14,18,19,29,30,36,45,46,48,50–54,59,61–64]</td>
<td></td>
</tr>
<tr>
<td>- Preparing an initial list of synonyms to search for</td>
<td>20</td>
</tr>
<tr>
<td>[14,18,19,29,30,36,45,46,48,50–54,59,61–64]</td>
<td></td>
</tr>
<tr>
<td>- Employing an iterative approach</td>
<td>13</td>
</tr>
<tr>
<td>- Sets were reviewed</td>
<td>26</td>
</tr>
<tr>
<td>- With input from a clinician</td>
<td>20</td>
</tr>
<tr>
<td>- Without input from a clinician</td>
<td>6</td>
</tr>
<tr>
<td>[51,54–56,60,64]</td>
<td></td>
</tr>
<tr>
<td>- Sensitivity/specificity discussion</td>
<td>19</td>
</tr>
<tr>
<td>[18–20,29,34,36,45,47–49,53–56,58–61,65]</td>
<td></td>
</tr>
<tr>
<td>- Calls for openness and sharing of code sets</td>
<td>14</td>
</tr>
<tr>
<td>[14–18,20,29,34,36,45,46,50,54,64,65]</td>
<td></td>
</tr>
<tr>
<td>- Suggesting a code set sharing approach</td>
<td>8</td>
</tr>
<tr>
<td>[19,29,31,45,51,59,63,64]</td>
<td></td>
</tr>
<tr>
<td>- The need for caution due to temporal and dynamic nature of code sets</td>
<td>13</td>
</tr>
<tr>
<td>[18,19,29,34,46,50,52,54–58,63,65]</td>
<td></td>
</tr>
<tr>
<td>- Described software or code to support code set selection</td>
<td>7</td>
</tr>
<tr>
<td>[18,19,29,58,60–62]</td>
<td></td>
</tr>
<tr>
<td>- Suggestions for tools</td>
<td>2</td>
</tr>
<tr>
<td>[36,50]</td>
<td></td>
</tr>
<tr>
<td>- Data driven methods</td>
<td>3</td>
</tr>
<tr>
<td>[58–60]</td>
<td></td>
</tr>
</tbody>
</table>

**Table 3**

Terminologies used in the methodological papers (eight papers used more than one terminology).

<table>
<thead>
<tr>
<th>Terminology</th>
<th>Medical coverage</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read v2</td>
<td>Comprehensive (diagnoses, signs, symptoms, procedures, laboratory tests, medications)</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>[18–20,29,34,45–49,53–56,58–61,63]</td>
<td></td>
</tr>
<tr>
<td>SNOMED</td>
<td>Comprehensive</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>[14,30,31,36,50,53,63]</td>
<td></td>
</tr>
<tr>
<td>ICD9</td>
<td>Diseases, signs and symptoms</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>[14,31,36,57,60,65]</td>
<td></td>
</tr>
<tr>
<td>ICD10</td>
<td>Diseases, signs and symptoms</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>[14,31,36,48,61]</td>
<td></td>
</tr>
<tr>
<td>LOINC</td>
<td>Laboratory observations</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>[31,36,50]</td>
<td></td>
</tr>
<tr>
<td>RxNORM</td>
<td>Medications</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>[31,36,50,64]</td>
<td></td>
</tr>
<tr>
<td>UMLS</td>
<td>Comprehensive</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>[50,51]</td>
<td></td>
</tr>
<tr>
<td>CPT</td>
<td>Surgical and medical procedures</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>[57]</td>
<td></td>
</tr>
<tr>
<td>CVT3</td>
<td>Comprehensive</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>[63]</td>
<td></td>
</tr>
<tr>
<td>OPCS4</td>
<td>Comprehensive</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>[61]</td>
<td></td>
</tr>
</tbody>
</table>

---

systems. All the distributed systems were, by definition, designed for a multi-site architecture. Just over half (n = 16) of the reviewed systems provided any detail about the code set construction process involved: a list of synonyms was created (n = 15) and then the hierarchy of a terminology utilised for searching (n = 14), with one additional paper using a library of pre-existing code sets [69]. The studies investigated and their characteristics are recorded in Appendix A.

The level of abstraction presented to the user varied across the different systems. Firstly, there is the level to which the systems attempted to translate coded records into a consistent format. One extreme is when during the system setup, coded concepts are translated into phenotypes with labels such as “diabetes” or “ace inhibitor”, and user interaction is limited to selecting these terms from a user interface component such as a drop-down list. In this model the act of constructing code sets is a one-time operation and abstracted away from end users. Systems designed like this include: Electronic Health Records for Clinical Research (EHR4CR) [70], Harvest [71] and the Health Care Systems Research Network [72]. The other extreme is to keep all coded records in the terminology in which they were originally coded, even if this is a non-standard local terminology, and then the act of creating the phenotype and the construction of code sets is left in its entirety to the end user, such as with 12b2 [38] and eMERGE [37]. In the middle systems like PCORnet [73], require that all local terms are mapped to a standard vocabulary or terminology, with construction of code sets limited to the mapped vocabulary.

Where systems allowed the sharing of information, this was always done by sharing the entire phenotype definition, rather than the code set in isolation.
5. Discussion

Given a completely error free terminology modelled as an ontology, and accurate clinical coding practices, there would perhaps be no need for the laborious process of constructing a code set. A researcher interested in type 2 diabetes would simply select all codes that were instances of that condition and they would be done. In practice this is not the case. Firstly, a terminology that does not support multiple inheritance will see codes for a condition occurring in separate branches of the tree requiring the researcher to exhaustively search across all terms without any form of filtering. Secondly, inaccurate coding practices necessitate that a researcher must also consider codes indicative of a covariate, for example using a code for “hypertension annual review” as a proxy for a patient with hypertension in the absence of a diagnostic code. Thirdly, terminologies may provide multiple ways to code the same concept. SNOMED is a good example of this where post-coordination allows, for example, at least four ways of describing acute appendicitis [77]. This is beneficial to end users such as clinicians who can code information with their preferred method, however it is problematic for researchers when constructing code sets. Finally, any errors in the terminology whereby a code for a condition is in the wrong part of a tree again necessitate an exhaustive search. The larger the terminology, the greater the problem. The high proportion of papers from the UK using Read v2 codes identified by this review is perhaps indicative of that problem. ICD9, although used extensively around the world, is a relatively small terminology so the issue of creating code sets is not that great. Perhaps with the transition from ICD9 to ICD10 in the US we will start to see the issue of code set creation discussed more frequently in the literature, such as the comment from the creators of the e-phenotyping system Duke Enterprise Data Unified Content Explorer (DEDUCE) who noted that “given the large increase in the number of codes available using ICD-10-CM, we will have to consider if a different search and select interface may be more appropriate than how we structure presentation of ICD-9-CM descriptions” [78]. SNOMED, although not without errors [52,79], at least supports multiple inheritance, but it is unknown whether the limited number of papers found by this review reflects: the absence of the problem of code set management with this terminology; the relatively low adoption of SNOMED in EHR systems; or a deficiency of our search methodology.

When electronic phenotyping systems abstract away the necessity for an end user to manage clinical code sets by the use of mappings to a common terminology or data model such as PCORnet [73] or the Observational Medical Outcomes Partnership Common Data Model (OMOP CDM) [80] it does not remove the necessity for code set management. Instead it moves the responsibility upstream to the process where the source data is mapped to the common data model. The burden is removed from the end user, but the importance of good code set management processes remains and is perhaps heightened as any errors on inconsistencies in the mapping process will have an impact on all future analyses using the platform.

5.1. Code set management

We now consider the four main components of code set management (construction, validation, sharing and reuse) and provide general recommendations on the process, along with specific recommendations for the software tools that we believe are necessary to assist researchers in this field. The full list of recommendations appears in Table 4.

5.1.1. Code set construction

The creation of an initial list of synonyms which is then iteratively updated after searching through a code dictionary is a frequent and seemingly sensible approach. In addition, the ability to utilise the hierarchy present in most code dictionaries not only saves time for researchers by being able to select all descendants of a relevant code without selecting each individual code, but also provides insurance against missing terms from the synonym list as related codes in a hierarchy may suggest alternative terms to search for.

Recommendation #1: Software tools for the construction of code sets should:
1. make use of the hierarchy of clinical dictionaries to assist in the searching and selection of codes;
2. facilitate the initial construction of a list of synonyms; and
3. facilitate the various stages of the review process from the selection of the initial list of synonyms, to the review of the included and excluded codes;
3. facilitate an iterative process by suggesting additional synonyms based on codes discovered by searching the code hierarchy that are not found themselves directly.

There already exist several software tools designed to facilitate the construction of code sets, however their use is seldom reported, suggesting they are underused, or at least have been underused in the past. Potential barriers to their uptake might be: lack of awareness of their existence; ignorance of their necessity; or deficiencies in the tool themselves, either in functionality or that they are time consuming to use. The tools are also solely focussed on the construction of a code set, and do not easily support researchers who wish to validate, share, extend or update code sets, which we feel is an important part of the code set management process and would be invaluable to researchers involved in this field.

Recommendation #2: Software tools should:
1. be open source, publicly available and easily accessible;  
2. be backward compatible with code sets produced with earlier versions of the tool;  
3. have minimal setup to facilitate widespread adoption;  

5.1.2. Code set validation

There are several processes that are encompassed by code set validation. The process of internal validation typically with a clinician, as seen in many papers here reviewed, is when the final code set is examined to confirm that all the codes included are relevant to the concept of interest. This is an important step to reduce type I errors, where an incorrect code is wrongly included. The acts of creating a list of synonyms, using a code hierarchy and searching iteratively, although an important part of the code set construction process, are also beneficial at reducing type II errors, where a valid code is incorrectly omitted.

Recommendation #3: Where possible a clinician should help to create the initial list of synonyms and review the final code set  
Recommendation #4: Software tools should:
1. facilitate the various stages of the review process from the selection of the initial list of synonyms, to the review of the included and excluded codes; and  
2. make this process as simple and quick as possible – a key driver to the concept of interest. This is an important step to reduce type I errors, where an incorrect code is wrongly included. The acts of creating a list of synonyms, using a code hierarchy and searching iteratively, although an important part of the code set construction process, are also beneficial at reducing type II errors, where a valid code is incorrectly omitted.

Recommendation #5: Sensitivity analyses with multiple code sets should be performed.

Recommendation #6: Software tools should facilitate and encourage the creation of multiple sets of codes for sensitivity analyses.

A more complete way to validate a code set would be to use a gold standard measure such as a manual notes review. People with clinical expertise would review a set of patient notes, including unstructured data such as narrative text, to determine whether each patient has the condition or covariate of interest. This annotated dataset can then be used to validate code sets by evaluating their precision and recall. The precise details of such an approach, such as how many patients to analyse, how many reviewers to use, and how to determine and report inter-rater agreement is considered outside the scope of this paper.

However, due to the potentially sensitive nature of EHR data, and the possibility for patient identification if narrative text is examined, the free text required for a manual review is often extremely challenging, if not impossible, to obtain. Performing such a review is also a time consuming process, so it is unlikely that all applied studies could utilise this approach. However there is still a need for code set validation. With the appropriate sharing of code sets there is no reason why this validation couldn’t be done externally by researchers from different institutions against different sources of data.

5.1.3. Code set sharing

Despite the frequent calls for transparency and sharing of codes sets, and the existence of several platforms for that purpose, there is little evidence that anything substantive is occurring. The role of journals cannot be understated. The uptake of standards such as CONSORT [81], which has improved the reporting of randomised control trials [82,83], was driven by journals refusing to publish trials without a completed checklist. Currently there is no incentive to researchers to publish their own code sets so it is understandable why it does not occur. The RECORD statement [33] calls for the publication of such code sets, so perhaps the situation will improve soon.

Recommendation #7: Journals should insist that code sets for retrospective studies on routinely collected data are published alongside the article.

Given the need to maintain such code set repositories in perpetuity, it is recommended that sites such as github.com or bitbucket. org are used to store code sets and their associated metadata. Storing the code sets in a version control system such as git facilitates the versioning of such objects, while storing them in online repositories like the ones listed above enhances their longevity as these sites store most of the world’s current open source software. In addition, there is an argument for public research and healthcare organisations providing or underwriting ‘eternal resource locators’ for such entities.

Recommendation #8: Platforms for sharing should be discoverable, maintained indefinitely and support versioning.

Recommendation #9: Online code repositories such as github.com and bitbucket.org could be used to store code sets and their associated metadata. A common format for such repositories would need to be developed and adopted widely.

5.1.4. Code set reuse

Although sharing the final code set is to be encouraged, such as on sites such as PheKB [37] and Clinicalcodes.org [67], we feel it is insufficient to truly facilitate their sharing and reuse. If simply presented with a set of codes for a concept, or alternatively several sets from different sources which are all slightly different, without being sure of their provenance, a researcher is unable to review or trust the sets and must repeat many of the steps taken to construct the original sets to convince themselves of their validity. If in addition to the code set, certain pieces of metadata were also captured, such as the initial list of synonyms used to search for the codes, and any codes that were rejected, then a second researcher can more easily validate, reuse and extend the code set. The list of synonyms can be reviewed to allow the researcher to determine how thorough the code set is likely to be, and the excluded codes give an indication as to how closely aligned the code set is to the researcher’s current needs. It is also possible to imagine automated soft-
ware tools that can ingest this metadata to allow code sets to be updated and extended either when the code set is deemed to be out of date due to changes over time, or when future researchers need a similar but non-identical code set.

**Recommendation #10**: Platforms for sharing should support the storage of metadata alongside the code set.

**Recommendation #11**: Software tools for constructing code sets should:
1. record metadata such as: the initial list of synonyms, the excluded codes, the purpose for the set and the author – standards such as the HL7 FHIR ValueSet schema could be reused or extended for this purpose; and
2. facilitate the reuse, validation and sharing of codes sets, not simply their construction.

5.1.5. Code set terminology

The different ways that people refer to code sets (code lists, value sets, etc.), makes it hard to ensure that researchers are comparing like with like, and acts as a barrier for people interested in the latest methodologies in this area. We have referred to these objects as clinical code sets, or just code sets, throughout this review and encourage other researchers to adopt this terminology. We feel this is the most accurate and unambiguous term and better than “code list” as a “list” implies an ordering which is not the case, and “value set” which more generally means a set of possible values for a given question/slot and is not specific enough.

**Recommendation #12**: The objects reviewed in this paper should be called “clinical code sets” or simply “code sets”.

5.2. Limitations

The review only searched for methodological papers on code set management. We did not search the huge volume of applied papers that have re-used EHR data for research so there may be documented methods for constructing code sets not included here. However, a recent review discovered that only 5% of applied papers share their code sets [19], so it is reasonable to assume that an even smaller proportion give sufficient detail as to their methodology so as to include them in this review.

The literature search and subsequent concept extraction was performed by a single author (RW) which may have introduced bias, however as the objective of the paper was a methodological review and encourage other researchers to adopt this terminology. We have presented a list of recommendations for the management of code sets and associated software tools.

For research conducted on EHR data it is insufficient merely to share final code sets used in analyses. Instead, the associated metadata explaining the method for constructing the code set should be shared alongside the set to make it transparent, reusable and reproducible. To do so will not only improve the quality of real world clinical studies from single healthcare systems but also facilitate a much needed scaling-up of research across heterogeneous populations, risk-environments and care-settings [84].

**Conflict of interest**

The authors declare that there are no known conflicts of interest.

**Acknowledgements**

Funded by the National Institute for Health Research Greater Manchester Primary Care Patient Safety Translational Research Centre (NIHR GM PSTRC). The views expressed are those of the author(s) and not necessarily those of the NHS, the NIHR or the Department of Health. MRC Health eResearch Centre grant MR/K006665/1 supported the time and facilities of EK, IB and NP.

### Appendix A. e-phenotyping systems reviewed

<table>
<thead>
<tr>
<th>System</th>
<th>Architecture</th>
<th>Multi site?</th>
<th>Synonym list</th>
<th>Hierarchy traversal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Screening for Active Protocols (ASAP)</td>
<td>Data warehouse</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Biomedical Translational Research Information System and its de-identified query tool</td>
<td>Data warehouse</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>CICTR</td>
<td>Federated</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Columbia</td>
<td>Data warehouse</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>DISCERN</td>
<td>Data warehouse</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Duke Enterprise Data Unified Content Explorer (DEDUCE)</td>
<td>Federated</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Electronic Health Records for Clinical Research (EHR4CR)</td>
<td>Federated</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Electronic Medical Records and Genomics (eMERGE) Network and its Record Counter (eRC)</td>
<td>Federated</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Electronic Primary Care Research Network (ePCRN) Research Workbench</td>
<td>Federated</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

(continued on next page)
### Appendix B. Methodological papers


References


Publication 2: Term sets: A transparent and reproducible representation of clinical code sets.
RESEARCH ARTICLE

Term sets: A transparent and reproducible representation of clinical code sets

Richard Williams¹,²*, Benjamin Brown¹,²,³, Evan Kontopantelis², Tjeerd van Staa², Niels Peek¹,²

1. Greater Manchester Patient Safety Translational Research Centre, University of Manchester, Manchester, United Kingdom, 2. Division of Informatics, Imaging and Data Science, The University of Manchester, Manchester, United Kingdom, 3. Centre for Primary Care, Division of Population Health, Health Services Research and Primary Care, The University of Manchester, Manchester, United Kingdom

* rrichard.williams@manchester.ac.uk

Abstract

Objective

Clinical code sets are vital to research using routinely-collected electronic healthcare data. Existing code set engineering methods pose significant limitations when considering reproducible research. To improve the transparency and reusability of research, these code sets must abide by FAIR principles; this is not currently happening. We propose ‘term sets’, an equivalent alternative to code sets that are findable, accessible, interoperable and reusable.

Materials and methods

We describe a new code set representation, consisting of natural language inclusion and exclusion terms (term sets), and explain its relationship to code sets. We formally prove that any code set has a corresponding term set. We demonstrate utility by searching for recently published code sets, representing them as term sets, and reporting on the number of inclusion and exclusion terms compared with the size of the code set.

Results

Thirty-one code sets from 20 papers covering diverse disease domains were converted into term sets. The term sets were on average 74% the size of their equivalent original code set. Four term sets were larger due to deficiencies in the original code sets.

Discussion

Term sets can concisely represent any code set. This may reduce barriers for examining and reusing code sets, which may accelerate research using healthcare databases. We have developed open-source software that supports researchers using term sets.
Conclusion

Term sets are independent of clinical code terminologies and therefore: enable reproducible research; are resistant to terminology changes; and are less error-prone as they are shorter than the equivalent code set.

Introduction

Clinical code terminologies, such as SNOMED [1] and ICD [2], are dictionaries of terms that allow clinicians to record events in electronic health records (EHRs) using alpha-numeric codes rather than free text. This makes patient records more manageable for clinical care, and allows secondary uses of the data, such as researchers performing retrospective observational studies. Researchers construct clinical codes sets [3–5] to represent the medical concepts they wish to investigate. This is a time-consuming activity, and prone to errors which can lead to biases in subsequent analyses [6]. Storing code sets in a format that facilitates validation, sharing and reuse is important, and called for frequently [7–10].

Code sets, also called code lists and value sets [5,9], range from one code to several thousand. The Value Set Authority Centre (VSAC) [11] provides a repository for code sets allowing their sharing and reuse. Their largest, for “Problem”, contains 117,930 SNOMED codes. This code set is likely not useful, but there are several that are and that contain thousands of codes: Trauma (ICD-10) 18524, Fracture lower body (ICD-10) 5902, Infection (SNOMED) 4066 and Cancer (SNOMED) 3867. Verifying large code sets, by checking that all included codes are correct, and also that no codes are missing, is an enormous task and acts as a barrier to reuse [3]. Updating code sets as terminologies change over time, and sub-setting or extending code sets, are laborious and error-prone activities.

This is important because differences in code sets can cause large variations in findings. Rodriguez et al [12] found rheumatoid arthritis (RA) incidence to be 0.15 per 1000 person-years, while Watson et al [13], in the same database, found it to be 1.03 per 1000 person-years; a sevenfold difference. Another study [14], calculated the weekly incidence of infectious intestinal disease as: 8.3/100,000 if using the World Health Organisation’s ICD-10 code set; 10.24/100,000 if using the Royal College of General Practitioners Research and Surveillance Centre’s ICD-9 code set; and 17.93/100,000 if using the ontological definition on which the paper was based.

The FAIR principles [15] aim to improve the transparency and reusability of scientific data and the algorithms and tools for processing and curating that data. Clinical code sets are a key part of the research process and should abide by FAIR principles; they should be findable, accessible, interoperable and reusable. This is not currently the case. Almost all code sets are unpublished [4] and therefore not accessible. Those that are published, on dedicated repositories such as VSAC or clinicalcodes.org [16], are findable but reuse is a challenge. In theory, reuse is achieved by downloading the relevant code set and applying it to an EHR database. However the task of checking the code set for errors involves reading the definition for each code to confirm that they are correctly in the set, and also speculatively searching the rest of the terminology for codes that may have been omitted. This is arguably as time-consuming as constructing the code set from scratch and is one of the current barriers to reuse. There is also no way currently to determine if a missing code was accidentally or deliberately omitted, therefore impossible to determine if a mistake was made, or if the code set definition contained a subtlety not otherwise described.
Objective

We propose a new representation of selection criteria for EHR based studies, based on lists of inclusion and exclusion terms. We introduce a methodology for constructing code sets which takes advantage of this representation, show that our method can represent any possible code set, and in doing so is typically more concise, and therefore practical for other researchers to verify, validate and ultimately reuse with confidence.

Materials and methods

We introduce ‘term sets’ to define cohort selection criteria for EHR-based studies. A ‘term set’ consists of three parts: inclusion terms describing the feature of interest (e.g. ‘stroke’, ‘heart failure’); exclusion terms describing things of no interest (e.g. ‘family history’, ‘screening’); and the target clinical code terminology and version (e.g. terminology = SNOMED-CT, version = uk-edition-v20180401). A code set is created from a term set by searching the terminology for codes that contain inclusion terms but that don’t contain exclusion terms.

Relationship between code sets and term sets

The traditional representations of cohort selection criteria are clinical code sets which are applied to EHR databases via a querying language. Code sets are extensional; they enumerate every code in the set. Term sets by contrast are intensional; they provide necessary and sufficient conditions by which a code is a member of the set. When applied to a particular terminology and version, a term set uniquely defines a code set. For example, consider the phrase “countries of the world” which is intensional, as compared with a complete list of countries of the world which is extensional. The list of countries changes over time, but at any point the intensional set can be derived from the extensional definition. Similarly, the extensional code set can be derived from the intensional term set.

Procedure for constructing term sets

Our method to construct a term set:

1. Select a clinical code terminology
2. Decide upon one or more inclusion terms, e.g. ‘heart failure’.
3. Perform a search within the terminology for codes with a definition matching the inclusion terms. The search rules are described below.
4. Optionally exclude matching definitions by adding exclusion terms. E.g. for ‘stroke’, it would make sense to exclude the term ‘family history’.
5. For hierarchical code terminologies, return codes that are descendants of matching codes, with definitions that do not contain an inclusion term. Add inclusion or exclusion terms to explicitly include or exclude these descendant codes.
6. Iterate until all inclusion terms have been added, and there are no unmatched descendants.

Deciding upon inclusion and exclusion terms is often a complex task requiring medical expertise. Therefore when implementing this method a clinician would need to be involved, or at the very least an expert in the particular disease domain. However for now we concentrate on the method itself, rather than its implementation. A worked example for the method can be found in S2 Appendix.
Search rules

Case insensitive. The term [fracture] matches “Shoulder fracture” and “Fracture of shoulder”.

Words are matched in any order. The term [shoulder fracture] matches “Shoulder fracture” and “Fracture of shoulder”.

All words must be present. The term [type 2 diabetes] matches “Diabetes, type 2” and “History of type 2 diabetes”, but not “Type 1 diabetes”.

Use quotes to match exactly. The term [“type 2 diabetes”] matches “Type 2 diabetes” and “History of type 2 diabetes” but not “Diabetes, type 2”.

Wildcards allow partial word searching. The term [diabet*] matches “Diabetes” and “Diabetic patient”.

Exact matches are never excluded. The term [heart failure] always matches “Heart failure” even if [heart] were excluded.

Proof that any code set can be represented as a term set

This ensures that our method can actually be used in practice for all code sets.

Clinical code terminology. A clinical code terminology \( T = (C,D,f) \) is a set of codes \( C \), a set of definitions \( D \), and a mapping function \( f : C \rightarrow D \) that links each code \( c \in C \) with a set of one or more definitions \( d \in D \). Examples for Snomed CT, Read v2 and ICD-10 would be:

\[
\begin{align*}
  f_{\text{snomedct}}(34486009) &= \{ \text{"Hyperthyroidism', 'Hyperthyroidism(disorder)' } \} \\
  f_{\text{readv2}}(G58..) &= \{ \text{"Heart failure', 'Cardiac failure' } \} \\
  f_{\text{icd-10(l71)}} &= \{ \text{"Rosacea' } \}
\end{align*}
\]

The mapping function is surjective; each element of \( D \) is mapped to by at least one element of \( C \). The inverse function \( f^{-1} : D \rightarrow C \) therefore exists for all definitions in \( D \) and is defined such that \( \forall d \in D, f^{-1}(d) = Y \) with \( c \in \{d \in f(c)\} \).

Matching definition set. For a set of word sequences \( W = (w_1, \ldots, w_m) \) and a terminology \( T = (C,D,f) \) we define the matching definition set \( MD(T,W) \) as the set of all definitions \( d \in D \) where \( w_i \) matches \( d \).

\[
MD(T, W) = \bigcup_{i=1}^{m} MD(T, w_i)
\]  

Matching definition set with exclusions. Given two sets of word sequences \( W,E \) and a terminology \( T = (C,D,f) \) we define the matching definition set with exclusions \( MDE(T,W,E) \) as the set of all definitions \( d \in D \) where \( w_i \) matches \( d \) and \( e_j \) does not match \( d \).

\[
MDE(T, W, E) = [W \cap D] \cup \{MD(T, W) \cap \{MD(T, E)\}^c\}
\]  

Matching concept set. For a terminology \( T = (C,D,f) \), and two sets of word sequences \( W, E \), we define the matching concept set \( M(T,W,E) \) as all codes in the terminology whose definition matches \( W \). Alternatively:

\[
M(T, W, E) = f^{-1}(MDE(T, W, E))
\]  

Proposal. Any subset of clinical codes from a terminology can be represented by a set of inclusion terms and a set of exclusion terms. Formally, for terminology \( T = (C,D,f) \) and any \( X = \{x_1,x_2, \ldots, x_n\} \), a subset of \( C \), there exists a set of inclusion word sequences \( I = \{i_1,i_2, \ldots, i_r\} \)
and a set of exclusion word sequences \( E = \{e_1, e_2, \ldots, e_r \} \) such that

\[
M(T, I, E) = X
\]

**Proof.** Let \( I = f(X) \) and \( E = f(X) \). Then

\[
M(T, I, E) = f^{-1}(MDE(T, I, E))
\]

From (3)

\[
= f^{-1}(I \cap D \cup [MD(T, I) \cap \{MD(T, f(X))\}^C])
\]

From (2)

\[
= f^{-1}([f(X) \cap D] \cup [MD(T, f(X)) \cap \{MD(T, f(X))\}^C])
\]

As \( I = f(X) \) and \( E = f(X) \)

\[
= f^{-1}(f(X) \cup \emptyset)
\]

As \( A \cap A^C = \emptyset \)

\[
= f^{-1}(f(X))
\]

\[
= X
\]

For a complete proof and all definitions, see S1 Appendix.

**Term set software**

We have developed a web application (https://getset.herokuapp.com) that implements the above methods and allows users to create and verify term sets. The tool is currently implemented for Read v2 codes (17) which are used in UK general practice, however it is straightforward to extend to other hierarchical terminologies like ICD or SNOMED. Once created, term sets can be automatically verified and then shared via GitHub (https://github.com/). Users are encouraged to add their name, a short title and description, so that researchers reusing their set can easily determine their intent.

**Empirical study**

The proof above demonstrates “completeness”; any code set can be represented as a term set. We also wished to demonstrate “efficiency”: a term set is shorter than the equivalent code set and is therefore easier and quicker to check. We therefore conducted an empirical study which found published clinical code sets, created their equivalent term set representations, and reported on their relative sizes.

GetSet is currently configured with Read v2, therefore we searched PubMed for papers using the Clinical Practice Research Datalink (CPRD) (18); a large primary care database containing Read v2 codes with 100s of publications annually. We used the search term (“CPRD”[all fields] or “Clinical Practice Research Datalink”[all fields]) and sorted the results by date descending. Reviewing recent papers ensured we can demonstrate that our method is valid for the current state of the art in clinical code set engineering.

We reviewed each paper in turn and included those that required the construction of code sets to define a cohort of patients. Cohort definition is the focal point of each paper and therefore the code set(s) that are most likely to appear. Also, by focusing on cohort definition, we avoided over-representation from papers with numerous code sets.

For each paper reviewed we extracted any code sets that described a patient cohort for a condition/diagnosis that had not been previously included. Certain conditions will likely be studied more frequently than others; restricting ourselves to one code set per condition ensured we had a sufficient variety of diseases.

We continued to review papers until code sets were discovered from 20 distinct papers. This ensured we would find 20 code sets for a variety of diagnoses and from a variety of authors.
We then created term set representations for each code set, using the above method, with the following caveats:

- Any ‘medcodes’ (CPRD’s code dictionary) were first converted to Read v2 codes.
- We removed all codes except Read v2 (e.g. CPRD also contains OXmis codes, which were in use pre-2000, and CTV3 codes).
- Where multiple codes have identical definitions, and the code set has included some but not all, we extended the code set to include them all.

For each code set we reported on the code set size and compared this with the number of inclusion and exclusion terms in our equivalent representation.

**Results**

The PubMed search was executed on 17th January 2018 by the lead author and returned 809 papers. The target of code sets from 20 distinct papers was reached after reviewing 45 papers; no further papers were reviewed. The 20 papers consisted of: 18 which included their code set in the paper, as a supplement, or in an online repository; 1 with code sets available on request so they were requested and received; and 1 that referenced code sets from another paper so this was retrieved to obtain the code sets. A total of 31 code sets for cohort definitions were found in the 20 papers. For further detail see: https://doi.org/10.5281/zenodo.1316984.

The median number of codes in each code set was 48 (IQR [18,120]). The smallest code set was for Stevens-Johnson syndrome and contained 1 code, while the largest code set, for infections that could lead to a potential hospitalization, contained 3,219 codes.

Each code set was successfully converted into a term set using our previously described procedure. The term sets are available at https://doi.org/10.5281/zenodo.1316984. The full list of code set definitions, their sizes, and the equivalent term set sizes are in Table 1. Nine code sets

Table 1. Codes set descriptions and sizes, the size of the related inclusion/exclusion term sets, and the inclusion/exclusion term sizes as proportions of the original code set size. Proportions ≤ 100% are displayed in bold.

<table>
<thead>
<tr>
<th>Cohort definition code sets</th>
<th>Code set size</th>
<th>Number of inclusion terms</th>
<th>Number of exclusion terms</th>
<th>Number of inclusion and exclusion terms as % of code set size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type 2 diabetes mellitus [19]</td>
<td>116</td>
<td>5</td>
<td>9</td>
<td>12.1%</td>
</tr>
<tr>
<td>Cancer except non-melanoma skin cancer [20]</td>
<td>1395</td>
<td>67</td>
<td>144</td>
<td>15.1%</td>
</tr>
<tr>
<td>Total knee replacement [21]</td>
<td>40</td>
<td>2</td>
<td>8</td>
<td>25%</td>
</tr>
<tr>
<td>Polymyalgia rheumatic [22]</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>33.3%</td>
</tr>
<tr>
<td>Asthma specific [23]</td>
<td>120</td>
<td>4</td>
<td>51</td>
<td>45.8%</td>
</tr>
<tr>
<td>Hidradenitis suppurativa (HS) [24]</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>50%</td>
</tr>
<tr>
<td>Shortness of breath excluded [2]</td>
<td>29</td>
<td>11</td>
<td>4</td>
<td>51.7%</td>
</tr>
<tr>
<td>Shortness of breath [2]</td>
<td>48</td>
<td>11</td>
<td>14</td>
<td>52.1%</td>
</tr>
<tr>
<td>Dementia [25]</td>
<td>74</td>
<td>8</td>
<td>31</td>
<td>52.7%</td>
</tr>
<tr>
<td>Non acute heart failure [26]</td>
<td>40</td>
<td>22</td>
<td>0</td>
<td>55%</td>
</tr>
<tr>
<td>Ethnicity [27]</td>
<td>183</td>
<td>46</td>
<td>63</td>
<td>59.6%</td>
</tr>
<tr>
<td>Potential hospitalized infections [28]</td>
<td>3219</td>
<td>1383</td>
<td>537</td>
<td>59.6%</td>
</tr>
<tr>
<td>Tuberculosis [29]</td>
<td>151</td>
<td>4</td>
<td>95</td>
<td>65.6%</td>
</tr>
<tr>
<td>Shoulder dislocation [30]</td>
<td>18</td>
<td>2</td>
<td>10</td>
<td>66.7%</td>
</tr>
<tr>
<td>Country of birth [32]</td>
<td>467</td>
<td>241</td>
<td>88</td>
<td>70.4%</td>
</tr>
<tr>
<td>Giant cell arteritis [33]</td>
<td>7</td>
<td>4</td>
<td>1</td>
<td>71.4%</td>
</tr>
</tbody>
</table>

(Continued)
Table 1. (Continued)

<table>
<thead>
<tr>
<th>Cohort definition code sets</th>
<th>Code set size</th>
<th>Number of inclusion terms</th>
<th>Number of exclusion terms</th>
<th>Number of inclusion and exclusion terms as % of code set size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type 1 diabetes mellitus [31]</td>
<td>35</td>
<td>4</td>
<td>25</td>
<td>82.9%</td>
</tr>
<tr>
<td>Psoriatic arthritis [32]</td>
<td>7</td>
<td>5</td>
<td>1</td>
<td>85.7%</td>
</tr>
<tr>
<td>Possible undiagnosed HS [24]</td>
<td>47</td>
<td>40</td>
<td>2</td>
<td>89.4%</td>
</tr>
<tr>
<td>Religion [27]</td>
<td>112</td>
<td>72</td>
<td>29</td>
<td>90.2%</td>
</tr>
<tr>
<td>Fragility fracture [33]</td>
<td>18</td>
<td>3</td>
<td>14</td>
<td>94.4%</td>
</tr>
<tr>
<td>Rheumatoid arthritis [34]</td>
<td>57</td>
<td>13</td>
<td>42</td>
<td>96.5%</td>
</tr>
<tr>
<td>Living alone [27]</td>
<td>65</td>
<td>39</td>
<td>25</td>
<td>98.5%</td>
</tr>
<tr>
<td>Colorectal cancer [35]</td>
<td>23</td>
<td>9</td>
<td>14</td>
<td>100%</td>
</tr>
<tr>
<td>Stevens-Johnson syndrome [36]</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Toxic epidermal necrolysis [28]</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>100%</td>
</tr>
<tr>
<td>Myotonic dystrophy type 1 [37]</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Marital status [27]</td>
<td>148</td>
<td>34</td>
<td>131</td>
<td>111.5%</td>
</tr>
<tr>
<td>Cohabitation [47]</td>
<td>85</td>
<td>22</td>
<td>79</td>
<td>118.8%</td>
</tr>
<tr>
<td>Residence [27]</td>
<td>168</td>
<td>92</td>
<td>111</td>
<td>120.1%</td>
</tr>
<tr>
<td>Heart failure [26]</td>
<td>55</td>
<td>20</td>
<td>48</td>
<td>123.6%</td>
</tr>
</tbody>
</table>

https://doi.org/10.1371/journal.pone.0212291.t001

omitted codes with definitions identical to an included code and so these codes were added prior to the conversion process. As an example, the code set for rheumatoid arthritis included the code “N040R00: Rheumatoid nodule”, but did not include the code “N042200: Rheumatoid nodule”, therefore N042200 was added prior to the conversion to a term set. The full list of extra codes for these nine code sets is available in S1 Table.

The total size of the term sets was on average 74% of the size of the code sets. In four code sets the total number of inclusion and exclusion terms exceeded the size of the code set: marital status, cohabitation, residence and heart failure. The code sets for marital status and cohabitation both use the code “1331.00: Single”. The inclusion term “single” matches many unrelated codes therefore many exclusion terms are needed. The code sets for residence and heart failure were perhaps poorly defined by the original authors. The residence code set aims to include codes that describe a person’s residential status and includes such wonderful terms as “Fall from cliff, occurrence in residential institution” and "Bitten by crocodile, occurrence in residential institution", but then doesn’t include the terms "Prolonged stay in weightless environment, occurrence in residential institution" or “Victim of avalanche, occurrence in residential institution”. In order to represent this precisely with a term set we needed to include a large number of unnecessary exclusion terms. Finally the heart failure code set includes some, but not all, cardiomyopathy codes. There is no clinical reason for this and the number of inclusion terms would reduce if “cardiomyopathy” could be included, as opposed to the current situation where the exact definition of 15 cardiomyopathy codes must be included.

Discussion

We have developed a method for creating clinical code sets that incorporates metadata on how the code set was created. We have demonstrated with a formal proof that our method works for any code set, and have shown empirically that the lists of inclusion and exclusion terms are on average shorter than the list of codes themselves.

A recent HL7 initiative provides a method for defining intensional value sets (code sets) [38]. Using this method a researcher can define a set of rules which when applied to a
terminology generate a code set. However this does not give the creator of the code set any support, methodology or tools for how to create the rules for the intensional definition. In a similar way, Reference Sets [39] within SNOMED can be used to specify a subset of concepts for use in a particular application, but without creation support. Reference sets are also specific to SNOMED. Our approach provides a generalizable methodology and software tool which are used to build term sets and their associated code sets. Integration of the approaches could be achieved if term sets created with our software were exportable to the HL7 definition of an intensional value set. This would then provide a robust and transparent code set creation process, along with a precise, formal definition.

There are at least four existing tools and associated methodologies for constructing clinical code sets. Davé and Petersen [40] created code sets by searching for synonymous terms and browsing the hierarchy. The final Stata script can be shared so that the process can be scrutinized. Others have developed R/Statascripts: pcdsearch [41] and CALIBERcodelists [42,43]. These scripts reuse the ideas of Davé and Petersen, while allowing more complex queries using Boolean operators and regular expressions. Recently Watson et al. [5] presented a three-stage process: defining the clinical concept a priori with clinician assistance; searching a clinical terminology using R or Stata to create an initial code set; and producing a final code set via a Delphi exercise with at least two GPs (the main difference to previous approaches).

Our approach builds on the strengths of these methods while addressing certain limitations. Each method above has a way of excluding codes; typically by specifying the codes themselves. By using exclusion terms, we produce metadata that is uncoupled from particular terminologies and is more readable to reviewers of the code set. The output of the above methods is always a script (Stata or R). By not tying our method to a particular scripting language, and using a simple web application, we reduce the barriers to methodical creation, inspection and reuse of code sets. Allowing regular expressions may help the code set creator, however it will likely act as a further barrier to reuse if the expressions get overcomplicated or if the next researcher is unfamiliar with regular expressions. We have kept our search strategy as simple as possible to mitigate this problem.

Although some of the reviewed code sets may have used one of the above methods, none made available the scripts used to create them. It is probably a safe assumption that this is true for the majority of code sets. The problem, for researchers reusing the code set, is that it is unknown which codes are missing and whether they were omitted deliberately or accidentally. Using our methodology these decisions become explicit. A future researcher may disagree with a decision, but at least it is available for scrutiny, and they can reuse the generated code set by tweaking the definition rather than starting from scratch.

Clinician involvement in code set development is critical, but precisely how research groups incorporate our methodology into their working practices is an open question. One option would be to use the three-stage process from Watson et al. with steps one and two (synonym definition and code set creation) facilitated with our tool.

We found examples where definitions only make sense when considered in the context of the hierarchy. E.g. the term "single" could be a numerical descriptor or a marital status. Our search strategy could be extended to examine the definitions of codes' ancestors. A search for "marital status single" would then return the code with the definition "single" only if it had ancestors that contained the words "marital" and "status". This would alleviate the problem where inclusion terms with low specificity ('single' as a marital status, 'white' as an ethnicity) lead to large numbers of exclusion terms.

The Read dictionary has a prefix-based hierarchy (G30's parent is G3, G3's parent is G). Two of the code sets we analysed (Dementia and potential hospitalized infections) used wildcards to represent multiple codes, e.g. "A" to represent "A..." and all of its descendants. This
leads to shorter code sets, which are easier to interpret, however it is problematic for two reasons. Firstly, when a code is included in a set it is not necessary that all descendants should also be included, and simply using a wildcard gives no guarantees that the researcher has inspected and accepted each code. Secondly, as the actual codes used in the analysis are not explicitly provided, it is impossible to determine which codes were actually used because code dictionaries change over time, with codes added and removed. Our methodology, which encourages users to specify inclusion (or exclusion) terms to match all descendants of included codes leads to more complete synonym lists and gives extra confidence to researchers reusing the code set.

Various problems were identified in the code sets (examples in Table 2). They fall into three categories: codes are omitted which do not correspond to the code set description; codes are omitted when they are obviously part of the code set; and some included and omitted codes are contradictory and should either all be included or all omitted. As we aimed to reproduce the code sets exactly, we have invariably created code sets with more inclusion and exclusion terms than are strictly necessary. By correcting the four code sets which had larger associated term sets we saw the average term set to code set proportion fall from 118.5% to 77.3%; all four term sets are now smaller than the code sets. For code sets constructed from scratch using our tool we would expect the number of inclusion and exclusion terms to be further reduced.

There are reasons why published code sets have omissions that aren’t necessarily errors. A researcher might justifiably decide that it is more important to capture a short list of codes which occur most frequently in their dataset than to focus on codes that occur infrequently or not at all. This may be true for their own research, but for other researchers wanting to reuse their code sets on different data sources it is not good enough. The burden of large code sets might have encouraged researchers to keep their code sets short, but with our methodology this is no longer a restriction, as validation can be performed on the shorter term sets rather than the code sets.

Another valid reason for omissions is that code dictionaries change over time so it is possible that codes recently added to a terminology do not appear in a code set. This becomes a question of how to best keep code sets updated over time, and our approach provides a simple way to do this. Previously when updating a code set a researcher, who hadn’t kept records of their search strategy from several years before, may end up recreating the code set. Now with the inclusion and exclusion terms captured and stored alongside the code set, one simply executes the term set definition against the updated code dictionary to see what additional codes may or may not need to be included.

We have demonstrated our method using Read codes, however the only precondition is that a terminology maps codes to definitions in a hierarchy, so our method would easily transfer to other terminologies such as SNOMED and ICD. One interesting avenue for further investigation is whether code sets can be translated into different terminologies. Once a researcher has defined a code set for one terminology, they could use the web tool to switch to a second terminology and automatically apply the same inclusion and exclusion terms to define a code set for that terminology. This would be useful for researchers using UK primary care data which is migrating from Read to SNOMED.

**Strengths**

We have shown that our method works formally via the proof and empirically via the code set mapping exercise. Using recent code sets from a variety of authors and for a variety of conditions demonstrates the generalisability of our technique. We have built upon the ideas from existing tools and methodologies as well as the recommendations from our earlier review [3].
Table 2. Examples of problems encountered with code sets.

<table>
<thead>
<tr>
<th>Code set</th>
<th>Example potential problems</th>
<th>Reason for problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frailty fracture</td>
<td>Included: S22.00 Fracture of humerus</td>
<td>The included and omitted codes are contradictory. This leads to additional, unnecessary, inclusion and exclusion terms.</td>
</tr>
<tr>
<td></td>
<td>S222000 Closed fracture of humerus NOS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Omitted: S22z.00 Fracture of humerus NOS</td>
<td></td>
</tr>
<tr>
<td>Potential hospitalized infections</td>
<td>Included: A53.00 Herpes zoster</td>
<td>The included and omitted codes are contradictory. This leads to additional, unnecessary, inclusion and exclusion terms.</td>
</tr>
<tr>
<td></td>
<td>F501611 Herpes zoster—otitis externa</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A35.00 Erysipelas</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Omitted: F501411 Erysipelas—otitis externa</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A8...00 Mycoses (and all descendant codes)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Omitted: FyuN500 Otitis externa in mycoses</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hyu8E00 Pneumonia in mycoses classified elsewhere</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N016.00 Arthropathy associated with mycoses</td>
<td></td>
</tr>
<tr>
<td>Type II diabetes mellitus</td>
<td>Included: C105100 Diabetes mellitus, adult onset, + ophthalmic manifestation</td>
<td>There is no clinical reason for type II diabetes why you would include the first two codes and exclude the second two. They should all be included.</td>
</tr>
<tr>
<td></td>
<td>C102100 Diabetes mellitus, adult onset, + unspecified complication</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Omitted: C100100 Diabetes mellitus, adult onset, no mention of complication</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C101100 Diabetes mellitus, adult onset, with ketoacidosis</td>
<td></td>
</tr>
<tr>
<td>Type I diabetes mellitus</td>
<td>Included: C10E00 Type I diabetes mellitus with hypoglycaemic coma</td>
<td>There is no clinical reason for type I diabetes why you would include the first two codes and exclude the second two. They should all be included.</td>
</tr>
<tr>
<td></td>
<td>C10E100 Type I diabetes mellitus with ketosidotic coma</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Omitted: C10E300 Type I diabetes mellitus with multiple complications</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C10E400 Type I diabetes mellitus without complication</td>
<td></td>
</tr>
<tr>
<td>Rheumatoid arthritis</td>
<td>Included: N065.00 Unspecified polyarthropathy or polyarthritis</td>
<td>The included and omitted codes are contradictory. This leads to additional, unnecessary, inclusion and exclusion terms.</td>
</tr>
<tr>
<td></td>
<td>Omitted: N065.11 Polyarthropathy not elsewhere classified</td>
<td></td>
</tr>
<tr>
<td>Marital status</td>
<td>Included: I3JL300 Wife alive</td>
<td>The included and omitted codes are contradictory. This leads to additional, unnecessary, inclusion and exclusion terms.</td>
</tr>
<tr>
<td></td>
<td>Omitted: I3JL700 Husband alive</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Included: I3JL00 Health of spouse</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Omitted: I3JE.00 Lives with spouse</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Included: I3JD.00 Partner unemployed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Omitted: I3IZ400 Partner alive</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I3IZ500 Partner unwell</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I3IZ600 Partner well</td>
<td></td>
</tr>
<tr>
<td>Cohabitation</td>
<td>Included: I3JL300 Wife alive</td>
<td>The included and omitted codes are contradictory. This leads to additional, unnecessary, inclusion and exclusion terms.</td>
</tr>
<tr>
<td></td>
<td>Omitted: I3JL700 Husband alive</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Included: I3JL00 Health of spouse</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Omitted: I3JE.00 Lives with spouse</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I3JH.11 Spouse left home</td>
<td></td>
</tr>
<tr>
<td>Living alone</td>
<td>Included: I3FH.00 Lives with relatives</td>
<td>These codes are examples of living with someone and are therefore not examples of living alone.</td>
</tr>
<tr>
<td></td>
<td>I3J6.00 Lives with grandfather</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I3J8.00 Lives with grandmother</td>
<td></td>
</tr>
</tbody>
</table>

(Continued)
Table 2. (Continued)

<table>
<thead>
<tr>
<th>Code set</th>
<th>Example potential problems</th>
<th>Reason for problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Residence</td>
<td>Included: U10F100 Fall from cliff, occurrence in residential institution</td>
<td>The included codes indicate residence in a residential institution. The omitted codes are equivalent to this and should be included.</td>
</tr>
<tr>
<td></td>
<td>U128100 Bitten by crocodile or alligator, occurrence in residential institution</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Omitted: U182100 Prolonged stay in weightless environment, occurrence in residential inst....</td>
<td></td>
</tr>
<tr>
<td></td>
<td>U196100 Victim of avalanche, occurrence in residential institution</td>
<td></td>
</tr>
<tr>
<td>Religion</td>
<td>Omitted: 13yL.00 Tibetan Buddhist</td>
<td>These codes are examples of religions and should be included.</td>
</tr>
<tr>
<td></td>
<td>Omitted: 13yu.00 Coptic orthodox</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Omitted: 13aS.00 Wesleyan Methodist</td>
<td></td>
</tr>
<tr>
<td>Country of birth</td>
<td>Omitted: 13dL.00 Born in Isle of Man</td>
<td>These codes are indicative of country of birth and so should be included.</td>
</tr>
<tr>
<td></td>
<td>Omitted: 13du.00 Born in Paros Islands</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Omitted: 13dv.00 Born in Greenland</td>
<td></td>
</tr>
<tr>
<td>Ethnicity</td>
<td>Omitted: 9TC..00 Roma ethnic group</td>
<td>These codes are descriptive of ethnicity and so should be included.</td>
</tr>
<tr>
<td></td>
<td>Omitted: 9TC0.00 Bulgarian Roma</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Omitted: 9TC1.00 Czech Roma</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Included: 956.00 Indian</td>
<td>The included and omitted codes are contradictory. This leads to additional, unnecessary, inclusion and exclusion terms.</td>
</tr>
<tr>
<td></td>
<td>Omitted: 1347.00 Indian</td>
<td></td>
</tr>
<tr>
<td>Heart failure</td>
<td>Included: G55.00 Cardiomyopathy</td>
<td>There is no clinical reason for heart failure why you would include one code for cardiomyopathy but then exclude others. They should all be included.</td>
</tr>
<tr>
<td></td>
<td>Omitted: G552000 Dystrophic cardiomyopathy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Omitted: G558400 Amyloid cardiomyopathy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Omitted: G558.00 Cardiomyopathy in disease BC</td>
<td></td>
</tr>
<tr>
<td>Shortness of breath</td>
<td>Omitted: 173p.00 Breathlessness causing difficulty eating</td>
<td>This code is a synonym for shortness of breath and so should be included.</td>
</tr>
<tr>
<td>Tuberculosis</td>
<td>Included: Ayu1900 Military tuberculosis, unspecified</td>
<td>The included and omitted codes are contradictory. This leads to additional, unnecessary, inclusion and exclusion terms.</td>
</tr>
<tr>
<td></td>
<td>Omitted: Ayu1800 Other miliary tuberculosis</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Included: Ayu1300 Respiratory TB unspecified, no mention of bacteriological confirmation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Omitted: Ayu1100 Respiratory TB unspecified, confirmed bacteriologically and histologically</td>
<td></td>
</tr>
<tr>
<td>Cancer not non-melanoma skin cancer</td>
<td>Omitted: R305800 Malignant neoplasm of fourth metacarpal bone</td>
<td>These are types of cancer and should be included.</td>
</tr>
<tr>
<td></td>
<td>Omitted: Ryu8.00 Malignant neoplasm of thyroid and other endocrine glands</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Omitted: R640000 B-cell acute lymphoblastic leukaemia</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Omitted: R624.12 Haery cell leukaemia</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Omitted: R509.00 Malignant melanoma of eye</td>
<td></td>
</tr>
<tr>
<td>Asthma</td>
<td>Included: 6791000 Health education—asthma self management</td>
<td>The included and omitted codes are contradictory. This leads to additional, unnecessary, inclusion and exclusion terms.</td>
</tr>
<tr>
<td></td>
<td>Omitted: 6791.00 Health education—asthma</td>
<td></td>
</tr>
</tbody>
</table>

https://doi.org/10.1371/journal.pone.0212291.t002

**Limitations**

The search for papers was performed by a single author, however given the transparency of the search strategy the biggest risk is that a paper containing a code set has been incorrectly rejected. This would presumably be a random bias and not affect the results. The list of papers reviewed is also available for inspection at https://doi.org/10.5281/zenodo.1316984.

The decision to select code sets for the cohort definition, rather than for the outcomes or the confounders, could have affected the results. However we found code sets for a wide variety
of conditions and had few problems converting them into our format, so consider it likely that this would extend to other conditions.

Code sets can be represented in multiple ways, some of which will be easier to understand than others. Some researchers may therefore be able to produce ‘better’ term sets. This can also be seen as a strength, as researchers are more likely to use term sets that are more clearly defined, so these term sets will prevail at the expense of those that are harder to understand.

There may be occasions where it is unclear if a code should be included or not, for example if clinicians use the code in different ways. At present one solution is to create two or more term sets that either include or exclude the uncertain codes. These term sets would have slightly different inclusion and exclusion lists, and their associated description would highlight how sensitive or specific the term set was.

Finally, although largely terminology agnostic, on occasion the particular inclusion and exclusion terms are loosely tied to the terminology used. One extreme example in Read v2 is for the term “G21x00: . . . without congestive cardiac failure” which misspells the word “cardiac”. When selecting this code you would need an inclusion term of “cardiac failure” which could be confusing and is unlikely to work in other terminologies. This is, however, an infrequent occurrence.

Conclusion

We have developed a new representation of cohort selection criteria for EHR based studies, a term set, which consists of: inclusion and exclusion terms; and a clinical code terminology and version. We have described a method to create term sets and developed an open source web application that implements this procedure. We have shown that our representation is as expressive as clinical code sets, but more efficient. Finally, term sets are easier to share, inspect, and reuse, because they are independent of specific (versions of) clinical terminologies. We expect that this will benefit transparent and reproducible research with EHR data.

Supporting information

S1 Appendix. Full proof and definitions. Full formal proof and all definitions for the claim that a term set can represent any code set.

(SDOCX)

S1 Table. Inconsistent duplicated codes. Codes added to code sets where a code with an identical definition had been excluded.

(SDOCX)

S2 Appendix. Worked example. Step by step construction of a term set for Type 2 Diabetes.

(SDOCX)

Acknowledgments

This work was funded by the National Institute for Health Research (NIHR) Greater Manchester Patient Safety Translational Research Centre (NIHR Greater Manchester PSTRC). The views expressed are those of the authors and not necessarily those of the NHS, the NIHR or the Department of Health and Social Care.

Author Contributions

Conceptualization: Richard Williams.

Investigation: Richard Williams.
Methodology: Richard Williams, Evan Kontopantelis, Niels Peek.

Software: Richard Williams.

Writing – original draft: Richard Williams.

Writing – review & editing: Benjamin Brown, Evan Kontopantelis, Tjeerd van Staa, Niels Peek.

References

1. SNOMED International. Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT) [Internet].


S1 APPENDIX – FULL PROOF AND DEFINITIONS

Definitions and notation

Note on character sets
For simplicity the below assumes that all definitions in a terminology only contain characters in ASCII\[42\]. In practice this is likely true for most languages using the Latin alphabet, with the exception that there are no accented characters (e.g. é, ö, ç) in ASCII. The logic and proof below remains valid with any alphabet.

Definition – “clinical code” and “clinical code definition”
Let $A$ be the alphabet of printable ASCII characters i.e. those with a decimal value between 32 and 126 inclusive. A clinical code $c$ is a finite sequence of printable ASCII characters:

$$c = c_1c_2 ... c_m \quad \text{with } m < \infty \text{ and } c_i \in A \ \forall \ i \in [1, m]$$ (1)

A clinical code definition $d$ is defined equivalently:

$$d = d_1d_2 ... d_n \quad \text{with } n < \infty \text{ and } d_i \in A \ \forall \ i \in [1, n]$$ (2)

In reality clinical codes (hereafter simply called codes) are typically shorter than clinical code definitions (hereafter called definitions), and definitions are readable in a human language, while codes are not. However the following proofs do not require these restrictions.

Definition - “clinical code terminology”
A clinical code terminology $T = (C, D, f)$ is defined by a set of codes $C$, a set of definitions $D$, and a mapping function $f: C \rightarrow D$ that links each code $c \in C$ with a set of one or more definitions $d \in D$.

Examples for SNOMED CT, Read v2 and ICD-10 would be:

$$f_{\text{SNOMED CT}}(34486009) = \{ \text{‘Hyperthyroidism’, ‘Hyperthyroidism (disorder)’} \}$$

$$f_{\text{Read v2}}(G65.) = \{ \text{‘Transient cerebral ischaemia’, ‘Transient ischaemic attack’, ‘Drop attack’} \}$$

$$f_{\text{ICD-10}}(L71) = \{ \text{‘Rosacea’} \}$$
The definition for each code can have multiple synonymous definitions and potentially translations into other languages.

The mapping function is surjective, i.e. each element of $D$ is mapped to by at least one element of $C$. The inverse function $f^{-1}: D \rightarrow C$ therefore exists for all definitions in $D$ and is defined such that $\forall \ d \in D, \ f^{-1}(d) = Y$ with $c \in Y \Leftrightarrow d \in f(c)$.

**Notation**

Given a terminology $T = (C, D, f)$, and $X = \{x_1, x_2, \ldots, x_n\}$ a subset of $C$, then we define $f(X) = Y$ where $Y = \bigcup_{i=1}^{n} Y_i$ is the set of definitions such that $f(x_i) = Y_i$ for $i \in [1, n]$. That is:

$$f(X) = f\left(\bigcup_{i=1}^{n} \{x_i\}\right) = \bigcup_{i=1}^{n} f(x_i) = \bigcup_{i=1}^{n} Y_i = Y$$ (3)

For $Y = \{y_1, y_2, \ldots, y_m\}$ a subset of $D$ we can also define $f^{-1}(Y) = Z$ where $Z = \bigcup_{i=1}^{m} Z_i$ is the set of codes such that $f^{-1}(y_i) = Z_i$ for $i \in [1, m]$. That is:

$$f^{-1}(Y) = f^{-1}\left(\bigcup_{i=1}^{m} \{y_i\}\right) = \bigcup_{i=1}^{m} f^{-1}(y_i) = \bigcup_{i=1}^{m} Z_i = Z$$ (4)

**Assumption**

We assume that the inverse function is a one-to-one mapping. That is $\forall \ c \in C$ let $Y = f(c)$ then for $d \in D, \ f^{-1}(d) = \{c\} \Leftrightarrow d \in Y$.

**Lemma**

Given a terminology $T = (C, D, f)$, and a subset of $C$, $A = \{a_1, a_2, \ldots, a_r\}$ we show $f^{-1}(f(A)) = A$.

**Proof**

$$f^{-1}(f(A)) = f^{-1}\left(f\left(\bigcup_{i=1}^{r} \{a_i\}\right)\right)$$

$$= f^{-1}(\bigcup_{i=1}^{r} f(a_i)) \quad \text{From (3)}$$
\[ f^{-1}(\bigcup_{i=1}^{r} Y_i) \]

By letting \( f(a_i) = Y_i \)

\[ = \bigcup_{i=1}^{r} f^{-1}(Y_i) \quad \text{From (4)} \]

\[ = \bigcup_{i=1}^{r} f^{-1}(U_{y \in Y_i} y) \quad \text{From (4)} \]

\[ = \bigcup_{i=1}^{r} U_{y \in Y_i} f^{-1}(y) \quad \text{From (4)} \]

\[ = \bigcup_{i=1}^{r} U_{y \in Y_i} \{a_i\} \quad \text{From assumption 0} \]

\[ = \bigcup_{i=1}^{r} a_i \]

\[ = A \quad \blacksquare \]

**Definition – “word boundary”**

Given a definition \( d = d_1 d_2 ... d_n \), we define \( \alpha \) as the empty character at the start of the definition, and \( \omega \) as the empty character at the end of the definition. Our definition is now \( d = \alpha d_1 d_2 ... d_n \omega \). If \( N \) is the set of non-alphanumeric ASCII characters, then we define a word boundary as any character in the set \( N \cup \{\alpha, \omega\} \).

**Definition – “word” and “word sequence”**

Given a definition \( d = d_0 d_1 d_2 ... d_n d_{n+1} \) (\( d_0 = \alpha, d_{n+1} = \omega \)), we define a word \( w = w_1 w_2 ... w_m \) as an alphanumeric contiguous subsequence surrounded by word boundaries. Or formally: \( \exists j \in [0, n - m] \) s.t. \( \forall i \in [1, m] \) \( w_i = d_{j+i} \) is not a word boundary and \( d_j \) and \( d_{j+m+1} \) are both word boundaries.

The definition for a word sequence is the same except each \( w_i \), apart from \( w_0 \) and \( w_m \), can now be a word boundary. A word is therefore a word sequence that does not contain a word boundary.

**Notation**

Given a definition \( d \), let \( d^* \) be the set of all word sequences contained within \( d \).

**Definition - “matching” and “exact matching”**

We say a word sequence \( w \) matches a definition \( d \) \( \Leftrightarrow w \in d^* \). If \( w = d \) we say it is an exact match.
Definition - “matching definition set”
Given a word sequence $w$ and a terminology $T = (C, D, f)$ we define the matching definition set
$MD(T, w)$ as the set of all definitions $d \in D$ where $w$ matches $d$.

For a set of $m$ word sequences $W = \{w_1, \ldots, w_m\}$, we define:

$$MD(T, W) = \bigcup_{i=1}^{m} MD(T, w_i)$$

(5)

Definition - “matching definition set with exclusions”
Given two word sequences $w, e$ and a terminology $T = (C, D, f)$ we define the matching definition set
with exclusions $MDE(T, w, e)$ as the set of all definitions $d \in D$ where $w$ matches $d$ and $e$ does not
match $d$. If $w$ exactly matches $d$, then $d \in MDE(T, w, e)$ even if $e$ matches $d$.

$$MDE(T, w, e) = [w \cap D] \cup [MD(T, w) \cap \{MD(T, e)\}^C]$$

(6)

For a set of $m$ word sequences $W = \{w_1, \ldots, w_m\}$, and a set of $n$ exclusion word sequences $E =
\{e_1, \ldots, e_n\}$, we define:

$$MDE(T, W, E) = [W \cap D] \cup [MD(T, W) \cap \{MD(T, E)\}^C]$$

(7)

Definition - “matching concept set”
For a terminology $T = (C, D, f)$, and word sequences $w, e$, we define the matching concept set
$M(T, w, e)$ as all codes in the terminology whose definition matches $w$. Alternatively:

$$M(T, w, e) = f^{-1}(MDE(T, w, e))$$

(8)

For a set of $m$ word sequences $W = \{w_1, \ldots, w_m\}$ we define:

$$M(T, W, E) = f^{-1}(MDE(T, W, E))$$

(9)
Main proposal

Proposal
Any subset of clinical codes taken from a terminology can be represented by a set of inclusion terms and a set of exclusion terms. Formally, given a terminology \( T = (C, D, f) \) and any \( X = \{x_1, x_2, \ldots, x_n\} \), a subset of \( C \), there exists a set of inclusion word sequences \( I = \{i_1, i_2, \ldots, i_r\} \) and a set of exclusion word sequences \( E = \{e_1, e_2, \ldots, e_s\} \) such that

\[
M(T, I, E) = X
\]  
(10)

Proof
Let \( I = f(X) \) and \( E = f(X) \). Then

\[
M(T, I, E) = f^{-1}(MD\{I, E\})
\]
From (9)

\[
= f^{-1}\left([I \cap D] \cup [MD(T, I) \cap (MD(T, E))^C]\right)
\]
From (7)

\[
= f^{-1}\left([f(X) \cap D] \cup [MD(T, f(X)) \cap (MD(T, f(X)))^C]\right)
\]
As \( I = f(X) \) and \( E = f(X) \)

\[
= f^{-1}\left(f(X) \cup [MD(T, f(X)) \cap (MD(T, f(X)))^C]\right)
\]
As \( f(X) \subseteq D \)

\[
= f^{-1}(f(X) \cup \emptyset)
\]
As \( A \cap A^C = \emptyset \)

\[
= f^{-1}(f(X))
\]

\[
= X
\]
\(\square\)
From lemma 0

Assumption revisited
If we drop the assumption that \( f^{-1}: D \rightarrow C \) is a one-to-one function then we no longer have \( f^{-1}(f(X)) = X \) from lemma 0. Instead \( f^{-1}(f(X)) = X \cup Y \) where \( y \in Y \) are codes not in \( X \) but with a definition identical to a code in \( X \). Therefore our approach does not work \( \forall X \subseteq C \) such that \( f^{-1}(f(X)) = X \cup Y \) where \( Y \neq \emptyset \). Or, less formally, if two codes have identical definitions then it is impossible to have a code set including one code, but excluding the other.
We would argue that a terminology should not have multiple codes with the same definition. However it does happen such as in Read v2 where 177..00, SM7y200 and SM7z.11 all have the definition ‘Smoke inhalation’, and N064K11, N094513, N096512, N220Y00 all have the definition ‘Irritable hip’. Given two distinct codes, x and y, with identical definitions there are two possible scenarios:

1. x and y represent identical concepts and should therefore either both be in a code set, or both be absent from a code set.
2. x and y represent non-identical concepts, in which case the terminology is at fault and the definitions should be changed to ensure that users of the terminology can distinguish between these two concepts.
Publication 3: Making medication data meaningful: Illustrated with hypertension.
Making Medication Data Meaningful: Illustrated with Hypertension

Richard WILLIAMS\textsuperscript{a,b,1}, Benjamin BROWN\textsuperscript{a}, Niels PEEK\textsuperscript{a,b} and Iain BUCHAN\textsuperscript{a,b}

\textsuperscript{a} MRC Health eResearch Centre

\textsuperscript{b} NIHR Greater Manchester Primary Care Patient Safety Translational Research Centre, University of Manchester, Manchester, UK

\textbf{Abstract.} We demonstrate, with application to hypertension management, an algorithm for reconstructing therapeutic decisions from electronic primary care medication prescribing records. These decisions concern the initiation, termination and alteration of therapy, and have further utility in: monitoring patient adherence to medication; care pathway analysis including process mining; advanced phenotype construction; audit and feedback; and in measuring care quality.

\textbf{Keywords.} Pharmacoepidemiology; phenotype extraction; drug prescriptions; electronic prescribing; care pathway.

\section{1. Introduction}

Electronic health records (EHRs) typically contain coded data about prescriptions, which, in the UK, are readily available for research from anonymised collections of primary care records [1, 2]. These data usually describe orders to dispense medication, but do not explicitly record the therapeutic decisions of when treatment is commenced, changed or terminated. These events, which are more clinically meaningful than individual prescriptions, can be used in a variety of analyses and tools: monitoring patient adherence to medication; care pathway analysis including process mining; next-generation phenotyping[3]; realistically-complex quality indicators; and advanced audit. However the raw EHR data must first be processed with consideration given to: the clinical codes; the drug family and active ingredients; and the dose amount, frequency and duration of all repeat prescriptions in a patient’s history. This pre-processing is often done as part of research but is usually simplified or tailored to the analysis [4–8].

A common approach for inferring drug usage is to count the number of prescriptions within the study period and to only include patients who exceed a certain threshold. Sometimes this is a single prescription [4], but more often this is two or more prescriptions [5–7], presumably to ensure a certain degree of continued usage and to exclude one off prescriptions. Another method for determining continued drug usage is to look for prescriptions in adjacent time windows [8]. These approaches are valid if the dosage or presence of a drug is considered as a single covariate, however, to contextualize prescribing to the clinical decisions made on care pathways it is necessary to convert the EHR data into decision events such as when therapy is commenced, changed or terminated.

\footnote{1 Corresponding Author: richard.williams2@manchester.ac.uk}
Similar research has been performed in extracting medication information from free text in discharge summaries [9, 10], from physician notes [11] and from drug purchase databases [12]. However, our approach relies on well coded prescription events, such as those available from the UK primary care system where all prescribing is electronic.

Tanskanen et al. [12, 13] construct drug use periods defined by a start and end date with an algorithm similar to the one described here. Our method goes further by detecting not only when therapy is commenced and terminated, but also when the dosage is changed leading to the extraction of a greater number of clinically meaningful events on a patient’s individual care pathway.

Here we focus on medications prescribed for hypertension.

2. Method

We used an anonymized extract of primary care data from 53 general practices in Salford, UK (population 234k) from the Salford Integrated Record (SIR). SIR collects primary care data for research purposes, with consent on an opt-out basis, from the general practices in Salford. The data consist of Read codes (version 2) and EHR vendor-specific codes. SIR contains coded data for all prescribing in primary care for its population for at least the past fifteen years.

All prescriptions of drugs recommended by the National Institute for Health and Care Excellence (NICE) for the treatment of hypertension [14] in the UK were extracted. These are angiotensin-converting enzyme inhibitors (ACEIs); angiotensin-II receptor blockers (ARBs); alpha-adrenoceptor blockers (α-blockers); beta-adrenoceptor blockers (β-blockers); calcium channel blockers (CCBs); thiazides and related diuretics; spironolactone and other diuretics. We extracted the clinical code, the number of tablets prescribed and the patient instructions (e.g. “Take 2 once a day”).

We created a mapping between each drug code, the active ingredient(s) and the tablet dose (mg). We found 653 Read codes and 199 vendor-specific codes, covering 178 brand names and 70 generic names of antihypertensive medications. This mapping information is publically available from the UK Health and Social Care Information Centre (HSCIC) for Read codes, however for the vendor-specific codes this was done manually based on the text description associated with each code.

Text mining, using regular expressions, on 216,101 distinct textual patient instructions yielded the number of tablets taken per day. We then iteratively developed an algorithm to take the amount, frequency, duration and type of medication, together with the prescription date to extrapolate meaningful events as shown in Table 1 and Table 2. The iterations continued until the proportion of unclassified events achieved an acceptably low level. The algorithm was then validated by two authors (RW, a software engineer, and BB, a clinician) who independently reviewed the records of a random sample of 100 patients to determine if the correct sequence of events had been extracted, and if not, recorded the discrepancy. Cohen’s κ showed fair inter-rater agreement (0.45). Disagreements were resolved through discussion.

The full algorithm including: code lists, mapping files, and regular expressions for converting patient instructions to tablets per day are all available online at https://github.com/rw251/research-events-medication-htn.
Table 1. The extracted medication events and the reasoning for each one

<table>
<thead>
<tr>
<th>Medication event</th>
<th>Reasoning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>Active ingredient first prescribed</td>
</tr>
<tr>
<td>Restart</td>
<td>Active ingredient previously prescribed but last event produced by the algorithm was a stop</td>
</tr>
<tr>
<td>Stop</td>
<td>Active ingredient previously prescribed but time has elapsed without a repeat prescription</td>
</tr>
<tr>
<td>Dose increase</td>
<td>Dose per day = (tablets per day) * (mg per tablet) increases</td>
</tr>
<tr>
<td>Dose decrease</td>
<td>Dose per day = (tablets per day) * (mg per tablet) decreases</td>
</tr>
</tbody>
</table>

Table 2. Example of a conversion from EHR to meaningful clinical events

<table>
<thead>
<tr>
<th>Date</th>
<th>EHR text</th>
<th>Instruction</th>
<th>Tabs</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-02-15</td>
<td>Atenolol 25mg capsules</td>
<td>1 in the am</td>
<td>28</td>
</tr>
<tr>
<td>2015-03-16</td>
<td>Atenolol 25mg capsules</td>
<td>Take 2 daily</td>
<td>28</td>
</tr>
<tr>
<td>2015-04-01</td>
<td>Atenolol 50mg capsules</td>
<td>One each day</td>
<td>28</td>
</tr>
</tbody>
</table>

3. Results

The algorithm was developed over six iterations involving the detailed examination of 179 patient records. A total of 10,311,973 prescriptions were extracted for 81,096 patients (demographic information in Table 3) over the period 7 July 1977 to 12 December 2014. The breakdown for each family of drugs is shown in Table 4. The algorithm produced sequences with a combined total of 850,028 events (28% starts, 34% stops, 15% restarts, 16% increases, 8% decreases and 0.02% unclassified).

Table 3. Patient characteristics for the 81,096 patients extracted from the dataset. Values are n (%) unless otherwise specified.

<table>
<thead>
<tr>
<th>Demographic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>62.43 (18.69)</td>
</tr>
<tr>
<td>Sex</td>
<td></td>
</tr>
<tr>
<td>Female</td>
<td>43419 (53.5%)</td>
</tr>
<tr>
<td>Male</td>
<td>37646 (46.4%)</td>
</tr>
<tr>
<td>Unknown</td>
<td>31 (0.04%)</td>
</tr>
<tr>
<td>Ethnicity</td>
<td></td>
</tr>
<tr>
<td>White</td>
<td>46712 (57.6%)</td>
</tr>
<tr>
<td>Other</td>
<td>4264 (5.26%)</td>
</tr>
<tr>
<td>Unknown</td>
<td>30120 (37.1%)</td>
</tr>
<tr>
<td>Deprivation (Index of Multiple Deprivation [15] quintiles)</td>
<td></td>
</tr>
<tr>
<td>1&lt;sup&gt;st&lt;/sup&gt; (Most deprived)</td>
<td>37850 (46.7%)</td>
</tr>
<tr>
<td>2&lt;sup&gt;nd&lt;/sup&gt;</td>
<td>19349 (23.9%)</td>
</tr>
<tr>
<td>3&lt;sup&gt;rd&lt;/sup&gt;</td>
<td>12262 (15.1%)</td>
</tr>
<tr>
<td>4&lt;sup&gt;th&lt;/sup&gt;</td>
<td>6851 (8.45%)</td>
</tr>
<tr>
<td>5&lt;sup&gt;th&lt;/sup&gt; (Least deprived)</td>
<td>3813 (4.70%)</td>
</tr>
<tr>
<td>Unknown</td>
<td>971 (1.20%)</td>
</tr>
</tbody>
</table>

Table 4. The number of antihypertensive drugs prescribed

<table>
<thead>
<tr>
<th>Drug family</th>
<th>Distinct drug types per family</th>
<th>Number of prescriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angiotensin-converting enzyme inhibitors</td>
<td>10</td>
<td>2,295,190</td>
</tr>
<tr>
<td>Angiotensin-II receptor blockers</td>
<td>7</td>
<td>841,217</td>
</tr>
<tr>
<td>Beta blockers</td>
<td>15</td>
<td>2,074,013</td>
</tr>
<tr>
<td>Calcium-channel blockers</td>
<td>11</td>
<td>2,021,822</td>
</tr>
<tr>
<td>Alpha blockers</td>
<td>8</td>
<td>543,226</td>
</tr>
<tr>
<td>Thiazide-like diuretic</td>
<td>11</td>
<td>1,278,005</td>
</tr>
<tr>
<td>Other diuretics</td>
<td>8</td>
<td>1,258,500</td>
</tr>
</tbody>
</table>
During validation the algorithm achieved a PPV of 92% (95% CI 85%-96%). Of the 100 records reviewed only eight had incorrect sequences and these were still partially correct. Four were missing a single stop event, two were false increases due to an erroneous prescription, one had an extra stop event, and one had a decrease for a switch from 2.5mg indapamide to 1.5mg modified release; these are actually clinically identical.

4. Discussion

We have developed a method for transforming unstructured and semi-structured prescription data into clinically meaningful therapeutic decisions on a care pathway from EHR data. From these events it is then easy to determine: when a patient is taking a medication; when there are adherence issues; when an intervention is or isn’t made by a physician; when guidelines are being followed correctly; and if treatment is having the desired effect. The events can also be used as part of a phenotype extraction process where the presence of a particular medication is indicative of a specific condition or diagnosis. Furthermore the algorithm has the potential to be improved by addressing the occasions where an incorrect sequence was produced.

The method described here addresses a specific aspect of the conversion of data from a primary care database into a form ready for further research and analysis. It can be viewed as a single module within a wider framework, where other modules might address problems such as constructing clinical code lists, imputing missing data or inferring diagnoses from other relevant information. This modular paradigm has many advantages over the monolithic approach: individual modules can be reused for multiple purposes; module development is more manageable and can be distributed across different research or analytic centres; and modules can be easily swapped for alternate or improved versions.

Future Work: We plan to incorporate these medication events into care pathways analyses to discover how pathway variation affects patient outcomes in managing long-term conditions. Strengths: The algorithm being developed by a software engineer and a clinician; both having extensive experience of primary care prescribing and primary care datasets; the iterative development of the algorithm; and the high PPV. Limitations: Development and validation were performed by the same authors; the algorithm is pharmacologically ignorant; we do not have information as to whether prescriptions were collected or taken. However, the act of prescribing is close to the clinician-centered decision context of care pathways.

Conclusion: Extracting research-quality information from routinely collected datasets is hard, time consuming, and prone to errors and bias. Accelerating research in these areas requires reproducible methods and tools that are open to scrutiny and easily reused and extended – all code for this project is available on github.com.

Acknowledgements

Funded by the National Institute for Health Research Greater Manchester Primary Care Patient Safety Translational Research Centre (NIHR GM PSTRC). The views expressed are those of the author(s) and not necessarily those of the NHS, the NIHR or the Department of Health.
References


Publication 4: Using String Metrics to Identify Patient Journeys through Care Pathways.
Using String Metrics to Identify Patient Journeys through Care Pathways

Richard Williams, BA¹², Iain E. Buchan, MD, FACMI¹², Mattia Prosperi, M.Eng., Ph.D¹, John Ainsworth, BSc, MSc¹²
¹Centre for Health Informatics, ²Greater Manchester Primary Care Patient Safety Translational Research Centre, University of Manchester, Manchester, UK.

Abstract

Given a computerized representation of a care pathway and an electronic record of a patient’s clinical journey, with potential omissions, insertions, discontinuities and reordering, we show that we can accurately match the journey to a particular route through the pathway by converting the problem into a string matching one. We discover that normalized string metrics lead to more unique pathway matches than non-normalized string metrics and should therefore be given preference when using these techniques.

Introduction

When faced with a patient’s electronic health record (EHR) and a prescribed care pathway it is useful to know if that patient’s care has deviated from the expected route through the pathway¹. The degree of deviation from a pathway calculated with a distance metric, when combined with outcome data, could lead to the discovery of instances where the standard of care has been suboptimal leading to adverse outcomes, and also to instances of localized practice that lead to better outcomes.

However, before determining distance from a given route, we need accurately to determine which route through the pathway was traversed by the patient. This is a problem because routinely collected patient information is often poorly recorded with missing data, incorrect coding practice and data recorded out of sequence.

String metrics provide the distance between two strings and are usually based on algorithms for matching strings to patterns, with various degrees of approximation. They typically involve performing operations such as insertion, deletion and substitution. The string metric can be normalized²,³ or non-normalized⁴–⁶.

We attempt to discover the routes patients took through a care pathway by using string matching methods in a novel way with electronic health records from Salford, UK.

Related Work

Representing a care pathway in a format that can be readily interpreted by a computer is essential for analysis and also enables health information systems to provide decision support to health care professionals⁷. Computer-interpretable guidelines (CIGs) are computer representations of the clinical knowledge in a clinical guideline and are usually networks of tasks that occur over time⁸. A recent review of CIGs shows there is ongoing work on CIG modelling languages, their integration with EHRs, validation and verification of CIGs, compliance monitoring and sharing⁹. Most CIG modelling is based on Task-Network Models⁸,⁹ of which our graph-based approach is a general case.

There is also a large body of work on process mining¹⁰,¹¹, frequent pattern mining, and the use of hidden Markov models for trajectory clustering¹² for healthcare data, which has been reviewed by Lakshmanan et al.¹³ However, each of these techniques begin with the healthcare data and attempts to interpolate the pathways taken, whereas our approach differs by starting with a well-defined care pathway and attempts to discover the route taken.

Background

Care Pathways

Care pathways are structured guidelines for the assessment, diagnosis, and treatment of patients with a given condition¹⁴–¹⁶. They provide the ideal care that a patient should receive and are often represented as a flow chart¹⁴. In the UK, “NICE Pathways” (National Institute for Health and Care Excellence) offers pathways for over 150 conditions¹⁷.

More formally, a care pathway flow chart can be represented as a directed graph, \( G = (V, E) \), with \( V \) a set of nodes that represent clinical events such as diagnoses, measurements, procedures and treatments, and \( E \) a set of directed edges that correspond to the permitted transitions between nodes. A transition can occur in a
determined amount of time. Figure 1 shows an example of a care pathway represented as a directed graph, defined a priori by experts.

**Figure 1:** A graphical model of a simplified, coded care pathway. Clinical codes in parentheses.

*SINAP*

The Stroke Improvement National Audit Programme (SINAP)\(^{18}\) is a data collection process for the purposes of clinical audit. It collects data about the care provided to stroke patients and includes several index events and the times they occurred. Here we examine data from Salford Royal Foundation Trust (SRFT) on 1078 patients with suspected strokes between 2010 and 2011. Figure 2 shows the approximate pathways that can be followed when a patient is admitted to hospital with a suspected stroke, covering the events recorded in the SINAP dataset. This is a simple pathway with only two decision points following when the patient is first seen and also after the patient has undergone brain imaging. The alphanumeric characters associated with each node in the pathway will be used later.

**Figure 2:** Stroke Improvement National Audit Programme (SINAP) pathway nodes as characters.

*Electronic Health Record*

A patient’s EHR is typically a list of coded events and states describing their care. In the UK a variety of coding schemes are used, such as Read Codes v2\(^{19}\), CTV3\(^{19}\), ICD-10\(^{20}\) and SNOMED\(^{21}\). The processes described in this paper can be used with any coding system: here we use the SINAP dataset that employs custom codes.
Method

Process

We first assign an alphanumeric character to each node in the graph. By using the Unicode\textsuperscript{22} character set we can manage care pathways with up to 65,536 nodes. We then extract every possible route through the pathway as a string made up of the characters assigned to each node. For a graph $G$ with $n$ possible routes we construct the set $R = \{R_1, R_2, \ldots, R_n\}$, where each $R_i$ is a string representing one of the $n$ possible routes. For acyclic graphs such as the stroke pathway for the SINAP dataset this is straightforward via recursion. For a directed graph with cycles it is possible to repeat a cycle indefinitely so the number of possible routes is infinite. To avoid this we only allow each cycle to be repeated a finite number of times.

Due to the nature of our data, the events recorded are all covered by the pathway. In general, however, when using records from primary or secondary care, they may not be consistent with a care pathway event/transition graph. For a single patient we therefore extract all timed events from their record that occur on the pathway of interest, convert the events to characters, and concatenate the characters into strings according to their date-time order. The strings then represent the patient’s journey through the care pathway.

If our dataset contains patients with multiple interactions with the pathway, we must then distinguish between distinct interactions with the care pathway by specifying a cut-off time. If ever the gap between adjacent patient events is greater than the cut-off, then we assume that the patient has left the pathway and any subsequent events form part of the patient’s next visit to the pathway. This works well when the timescale of a pathway is shorter than the distances between them.

We then use the following string metrics to determine the distance between a patient pathway and each possible route through a care pathway.

Longest Common Subsequence

Formally, given two sequences $A = a_1 a_2 \cdots a_m$ and $B = b_1 b_2 \cdots b_n \ (m \leq n)$ we say that $A$ is a subsequence of $B$ if there are indices $0 < j_1 < j_2 < \cdots < j_m \leq n$ such that $a_i = b_{j_i}$ is true for $i = 1, 2, \ldots, m$.

Given two sequences $X$ and $Y$, $Z$ is a common subsequence if it is a subsequence of both $X$ and $Y$. $Z$ is the longest common subsequence (LCS) if $|Z| \geq |Z'|$ for all common subsequences $Z'$, where $|X|$ is the length of $X$. The LCS is not necessarily unique.

We are interested in which route through the pathway a patient took so we need to decide on a distance metric to convert the LCS into something more meaningful. An initial algorithm for a single patient is as follows:

1. Create a list of all the possible routes $R_1, \ldots, R_n$ through the care pathway
2. Filter the patient’s events to just include pathway events and apply the time cut-off to give an event sequence $E = E_1 \ldots E_m$
3. For each route $R_i$ calculate $L_i = \text{LCS}(R_i, E)$
4. If $L_i > 0$ calculate the distance $d_i = \max(|R_i|, |E|) - |L_i|$
5. Return the set of routes with the smallest distance

However, this only considers the discrepancy between the LCS and the pathway route; it doesn’t take into account the length of the LCS. We can normalize the distance by either dividing by the LCS, or by dividing by the combined length of the two strings and step 4 above becomes either:

4. If $L_i > 0$ calculate the distance $d_i = \frac{\max(|R_i|, |E|) - |L_i|}{|L_i|}$

or

4. If $L_i > 0$ calculate the distance $d_i = \frac{\max(|R_i|, |E|) - |L_i|}{|R_i| + |E|}$

We call these two methods LCS1 and LCS2 respectively.

Simple Edit Distance (Levenshtein Distance)

An alternative to the LCS is to consider the edit distance or Levenshtein distance\textsuperscript{4}. The edit distance between two strings $X$ and $Y$ is the minimum number of operations required to convert $X$ into $Y$ where an operation is either: insert a character, delete a character or replace a character. When switching is allowed (\textit{ab} $\rightarrow$ \textit{ba}) the algorithm is the Damerau-Levenshtein\textsuperscript{5,6}. The costs of inserting, deleting and replacing are given as $W_I, W_D, W_R$. 

1210
and \( W_R \) respectively. It holds that \( W_R \leq W_D + W_I \) as we can always delete and then insert instead of substituting. By default the cost of each operation is 1.

The algorithm for our problem would be:

1. Create a list of all the possible routes \( R_1, \ldots, R_n \) through the care pathway
2. Filter the patient’s events to just include pathway events and apply the time cut-off to give an event sequence \( E = E_1 \ldots E_m \)
3. For each route \( R_i \) calculate the distance \( d_i = LEV(R_i, E) \)
4. Return the set of routes with the smallest distance

Similarly we can do this for the Damerau-Levenshtein distance which we will notate as \( d_i = DAM(R_i, E) \).

Levenshtein Variants

Several versions of the Levenshtein Distance normalized to the length of the strings have been suggested. We notate the following as \( NLEV^2 \).

\[
NLEV(X, Y) = \frac{LEV(X, Y)}{|X| + |Y|}
\]

Also a normalized Levenshtein distance that satisfies the triangle equality and is therefore a true distance metric:

\[
NLD(X, Y) = d_{N-GLD}(X, Y) = \frac{2 \cdot LEV(X, Y)}{\alpha \cdot (|X| + |Y|) + LEV(X, Y)}
\]

where \( \alpha \) is whichever cost is greater out of insertion and deletion\(^3\). However, when \( \alpha = 1 \), as is the case when all the weights are set to 1 by default, although the distances produced by NLD and NLEV will differ, the ordering of the matches will always be the same.

Finally, we consider a normalized version of the Damerau Levenshtein distance.

\[
NDAM(X, Y) = \frac{DAM(X, Y)}{|X| + |Y|}
\]

We compare and contrast the different distance measures: LCS1, LCS2, LEV, DAM, NLEV, NLD and NDAM.

Data cleaning

Right censoring of the data is unlikely as once in hospital all end points are recorded. Most times in the data seem to be rounded to the nearest 10 or 15 minutes. This may potentially result in events appearing simultaneously or even out of order. There is also a risk of recollection or estimation bias as the data is often captured after the event.

When events occur at the same time there are several options available. The patient can be ignored, but this would result in a lot of data being excluded from the analysis. An alternative would be to perform the analysis on the data ordered randomly and let the string matching methods correct any discrepancies. However as we are interested in discovering the actual path the patient took, we can assume where possible the events occurred in the correct order.

For two events A and B on a pathway there is either: a one-way path from A to B, a one-way path from B to A, a path from A to B and B to A, or it is impossible to get from one to the other. For a group of events occurring at the same time if it is possible to order them in a unique way then we choose that as the order of the events. If it is not possible, because of a cycle or an unreachable node, then we discard that patient. For datasets where this is commonplace it may be better to include the patients discarded here and randomise the order of the coterminaneous events. Alternatively we could just discard the events rather than the patient.

Similarly, events of unknown time, or those with just a date and not a time, can be inserted at the correct point of a patient record, if possible, or discarded if contradictions arise.

Data Management and Analysis Environment

The SINAP dataset was transferred to us via an encrypted external hard drive in CSV format. This was then uploaded to a Microsoft SQL Server 2008 database for analysis. Sequence matching was performed with
C#.NET and all statistical analysis was done using R\textsuperscript{23}. The sm library\textsuperscript{24} was used for plotting density curves and the pROC\textsuperscript{25} package was used for comparing Receiver Operating Characteristic (ROC) curves.

**Results**

*Data Characteristics*

The SINAP dataset contains 1078 patients of which 549 are female and 529 are male.

Table 1 shows the number of records that were cleaned using the above data cleaning process. Only 1 patient’s route could not be uniquely re-ordered.

**Table 1. Data cleaning results**

<table>
<thead>
<tr>
<th>Total patients</th>
<th>1078</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midnight events – able to insert</td>
<td>424</td>
</tr>
<tr>
<td>Simultaneous events – able to order</td>
<td>3</td>
</tr>
<tr>
<td>Midnight and simultaneous events – able to order</td>
<td>648</td>
</tr>
<tr>
<td>No midnight or simultaneous events – no need to order</td>
<td>2</td>
</tr>
<tr>
<td>Midnight events – unable to insert</td>
<td>1</td>
</tr>
</tbody>
</table>

There are 46 distinct pathways taken by the 1077 patients following time reordering. Table 2 shows the frequency of the top 10 patient pathways. The pathways that match the ICP are in bold. The route of GHDB should be a valid route however there are no patients in our cohort who followed this – suggesting this is not a valid route and the care pathway could be altered.

**Table 2. Top 10 pathways – character sequences from figure 2.**

<table>
<thead>
<tr>
<th>Patient Record</th>
<th>Count</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>GHDEFIB</td>
<td>275 (26%)</td>
<td>Valid route</td>
</tr>
<tr>
<td>GHDFIB</td>
<td>275 (26%)</td>
<td>Valid route</td>
</tr>
<tr>
<td>GHDFEIB</td>
<td>122 (11%)</td>
<td>Valid route with E/F switched – lots of people so maybe a valid route.</td>
</tr>
<tr>
<td>GDHEFIB</td>
<td>63 (6%)</td>
<td>Valid route with D/H switched – can’t be seen before you arrive.</td>
</tr>
<tr>
<td>GDHFIB</td>
<td>60 (6%)</td>
<td>Valid route with D/H switched – as above.</td>
</tr>
<tr>
<td>GHDEAFCIB</td>
<td>56 (5%)</td>
<td>Valid route</td>
</tr>
<tr>
<td>GHEDFIB</td>
<td>39 (4%)</td>
<td>Valid route with E/D switched – can’t be imaged before first seen.</td>
</tr>
<tr>
<td>GHDEACIB</td>
<td>37 (3%)</td>
<td>Valid with A/F switched.</td>
</tr>
<tr>
<td>GHDFIB</td>
<td>24 (2%)</td>
<td>D/F switched – can’t arrive in specialist bed before being seen.</td>
</tr>
<tr>
<td>GHDFEIB</td>
<td>24 (2%)</td>
<td>D/H and E/F switched</td>
</tr>
</tbody>
</table>

It appears that there are some valid routes that aren’t in our pathway. For those who don’t get thrombolysed there are many people who arrive in a specialist stroke bed prior to their brain scan. Also there are many people who get “First Seen” before they arrive at the hospital. This seems nonsensical but could be valid if “First Seen” applied to GPs or ambulance staff. Finally there are patients who receive thrombolysis after getting to a specialist stroke bed which could also be a valid route. All other switches appear to be mistakes – for example having a brain scan prior to being first seen.

In order to determine how well each method works we must determine for each patient the most probable route taken. As our dataset is small we can do this manually by defining rules based on the data. We first assume that events that don’t happen are rarely inserted and then classify the patients according to the following rules:

1. If a patient has thrombolysis or a follow up scan then assumes route GHDEAFCIB
2. Of those remaining, for any with a brain scan we assume route GHDEFIB
3. Of those remaining, for any with a stroke unit arrival or discharge we assume route GHDFIB
4. Of those remaining we assume GHDB

In addition to returning the correct result it is also of use if the distance measure returns a unique result. There will be situations where this isn’t possible but in general string matching methods that return more unique results are preferable.

For each method, Table 3 gives the number of unique matches and the number of correct matches where a correct match is one that is both unique and matches with the routes we assume the patients actually followed.
Table 3. Number of unique and correct matches

<table>
<thead>
<tr>
<th>Method</th>
<th>Unique Matches</th>
<th>Correct Matches</th>
<th>Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEV</td>
<td>818 (75.95%)</td>
<td>645 (78.85%)</td>
<td>59.89%</td>
</tr>
<tr>
<td>DAM</td>
<td>853 (79.20%)</td>
<td>849 (99.53%)</td>
<td>78.83%</td>
</tr>
<tr>
<td>LCS1</td>
<td>882 (81.89%)</td>
<td>878 (99.55%)</td>
<td>81.52%</td>
</tr>
<tr>
<td>LCS2</td>
<td>1077 (100.00%)</td>
<td>1070 (99.35%)</td>
<td>99.35%</td>
</tr>
<tr>
<td>NLEV</td>
<td>1076 (99.91%)</td>
<td>841 (78.16%)</td>
<td>78.09%</td>
</tr>
<tr>
<td>NLD</td>
<td>1076 (99.91%)</td>
<td>841 (78.16%)</td>
<td>78.09%</td>
</tr>
<tr>
<td>NDAM</td>
<td>1076 (99.91%)</td>
<td>1068 (99.26%)</td>
<td>99.16%</td>
</tr>
</tbody>
</table>

The NLEV and NLD methods produce the same results as predicted. The ratio of correct matches to unique matches shows that the Damerau-Levenshtein and the longest common subsequence methods work excellently with >99% correct, whereas the Levenshtein variants only achieve 78-79%. It can also be seen that normalized methods are better at producing unique matches with LCS2 matching all pathways uniquely, while NLEV, NLD and NDAM only fail to give a unique answer for a single patient - actually a different patient for each method. Examining the difference between NLEV and NDAM shows that NDAM is correctly identifying pathways where events have been recorded out of sequence. As an example the patient record of GHDFEIB is correctly matched to GHDEFIB by NDAM, while NLEV matches it to GHDFIB.

When the values for unique correct matches are combined the normalized Damerau-Levenshtein and the second Longest Common Subsequence methods are best, correctly matching >99% of the patient pathways.

For these two methods we can split the pathways into two groups: correct and incorrect matches, where a correct match is when the algorithm uniquely identifies the route the patient traversed through the pathway. We then compare the groups under the null hypothesis that the mean ‘string’ distance between them is equal. The density plots in Figure 3 demonstrate the data we want to contrast are not drawn from normal or symmetrical distributions, indeed the distributions of string distances are quite different for matches compared with non-matches. Thus we make the contrast with a non-parametric (Matt-Whitney) method, demonstrating statistically highly significant differences for both NDAM (P < 0.0001) and LCS2 (P < 0.0001) metrics.
Finally, we compare NDAM, LCS2 and NLEV string distance metrics with regard to their classification accuracy for our care pathway journeys. Figure 4 shows the ROC curves for each metric with our test dataset, and the 95% confidence intervals for the areas under the curves: the more detailed comparison of the two most accurate metrics (NDAM and LCS2) is the Mann-Whitney result above.
Figure 4: Receiver Operating Characteristic (ROC) curves for NDAM, NLEV and LCS2 string distance metrics with 95% confidence intervals for the areas under the curves.

Discussion

Distance Weighting

The operations in the Damerau-Levenshtein string metric can be weighted. Given the nature of our dataset it is more likely that records were omitted or out of order, than miscoded. If we are sure of this we can change the weighting of the operations accordingly – an option that is possible with the NDAM and not the LCS2 method. By doubling the weight associated with deleting a character, therefore making it less likely that matches will feature deletions, of the 1077 patients we yield 1077 unique matches of which 1074 are correct. Weighted NDAM then becomes the most accurate way of predicting a patient’s route.

Generalization

The string matching process described here operates on a graph based representation of a care pathway. Therefore the methodology is theoretically applicable, although untested, to any process or workflow that can be represented as a graph, in healthcare and beyond.

Future work

There are several factors unstudied in this paper that will affect the overall success of the method. The size and shape of the graph is a factor, as is the quality of the data. Further work is needed to determine which graph shapes work well with this method. Finally, the next stage of our work is to determine how the distance a patient is from their care pathway predicts their outcomes.

Conclusion

String matching would seem to be a highly successful way to determine which route a patient followed in a care pathway. Normalized distance functions should be used to ensure high numbers of unique matches. For clinical data where the chance of events occurring, or being recorded, in the wrong order is high, the Damerau-Levenshtein or Longest Common Subsequence methods should be used in preference to the Levenshtein distance.

Acknowledgements

Funded by the National Institute for Health Research Greater Manchester Primary Care Patient Safety Translational Research Centre (NIHR GM PSTRC). The views expressed are those of the author(s) and not necessarily those of the NHS, the NIHR or the Department of Health.
References


Process Mining in Primary Care: A Literature Review

Richard WILLIAMS\textsuperscript{a,b,1}, Eric ROJAS\textsuperscript{c}, Niels PEEK\textsuperscript{a,b} and Owen A JOHNSON\textsuperscript{d}
\textsuperscript{a}MRC Health eResearch Centre, Division of Informatics, Imaging and Data Science, University of Manchester, Manchester, UK
\textsuperscript{b}NIHR Greater Manchester Patient Safety Translational Research Centre, University of Manchester, Manchester, UK
\textsuperscript{c}Computer Science Department, School of Engineering, Pontificia Universidad Católica de Chile, Chile
\textsuperscript{d}School of Computing, University of Leeds, Leeds, UK

\textbf{Abstract.} Process mining is the discipline of discovering processes from event logs, checking the conformance of real world events to idealized processes, and ultimately finding ways to improve those processes. It was originally applied to business processes and has recently been applied to healthcare. It can reveal insights into clinical care pathways and inform the redesign of healthcare services. We reviewed the literature on process mining, to investigate the extent to which process mining has been applied to primary care, and to identify specific challenges that may arise in this setting. We identified 143 relevant papers, of which only a small minority (n=7) focused on primary care settings. Reported challenges included data quality (consistency and completeness of routinely collected data); selection of appropriate algorithms and tools; presentation of results; and utilization of results in real-world applications.

\textbf{Keywords.} Process mining; workflow; primary care; care pathways.

1. Introduction

Healthcare systems worldwide are trying to reduce costs by moving as much care as possible out of the hospital environment into other settings such as primary care. Primary care covers many aspects of healthcare, in a generalist manner, with a particular focus on the management of chronic conditions. In countries such as the UK where primary care plays a key role in the delivery of healthcare, patients are enrolled with a local, community-based general practitioner (GP) who is the first point of contact for non-emergency healthcare needs. Typically GPs act as the “gatekeeper” for referral to specialist care services and have responsibility for managing the lifelong care of the patient. Primary care is less structured than in-hospital care: patients are not physically present for the duration of their care pathway; care frequently transfers between healthcare settings and providers; and patients have a greater responsibility for the self-management and treatment of their conditions.

\footnotesize{1 Corresponding Author, Richard Williams, G.303a, Jean McFarlane Building, University of Manchester, M13 9PL; E-mail: richard.williams2@manchester.ac.uk.}
Process mining, also called “process discovery” or “workflow mining”, is the discipline of discovering processes from event logs, checking the conformance of real world events to idealized processes, and ultimately finding ways to improve those processes. It was originally applied to business processes over 20 years ago [1], and more recently has been applied to the healthcare domain. There are examples in secondary care [2, 3], tertiary care [4] and dentistry [5], but there would appear to be little published work in community or primary care.

Three recent literature reviews [6–8] related to process mining in healthcare have variously reported on: the volume of research over time; the algorithms and techniques used; the tools and software used; the geographical distribution of datasets and the medical domains studied. None of the reviews have specifically focused on the healthcare setting in which process mining was applied.

Therefore the objectives of this paper are: (1) to review the scientific literature on process mining in healthcare as it relates to community-based and primary care, (2) to summarize the data sources, geographical location and medical domains that were reported, and (3) identify challenges that may appear when applying process mining in primary care.

2. Method

We aimed to review articles written in English that are related to the application of process mining within healthcare and describe the use of a real world data source i.e. not simulated data or methods presented without data. A previous literature review [6] executed on 8th February 2016 used similar criteria, with the exception that they included papers with methods but no data. We therefore restricted our search to articles published since February 2016 until the present and included all the articles in the previous review that had a real world data source. Other literature reviews were considered out of scope on the grounds that they didn’t contain a data source.

Following [6], the databases searched were PubMed, dblp and Google Scholar. The Google Scholar searches were performed in an incognito mode to remove the effects of previous browser history. The searches were performed by the lead author on 5th October 2017 using the query: (“process mining” OR “workflow mining”) AND healthcare. Due to the domain specific nature of PubMed the “healthcare” part of the query was omitted for this database. We were careful to reproduce the same search strategies as [6] and this review also included the list of papers from the main process mining research community website at http://www.processmining.org/ so we also looked for papers here.

In total 579 papers were found after removing 31 duplicates. One was the previous literature review, 380 were excluded based on the title and 95 based on the abstract, leaving 103 to read in full. At this stage 73 of the 74 papers from the previous literature review were added after one duplicate had been removed. From the 176 papers read in full, 33 were excluded. For papers excluded at any stage, the most commonly used exclusion criteria were: not about healthcare (n=382); not process mining (n=67); and no data – just methods, discussion, literature review or simulation (n=24). This review is based on the 143 papers read, after duplicates and exclusions removed (see Figure 1).

The primary focus for data extraction was the healthcare setting of the datasets used for the process mining. The precise classification was deliberately left open ended to allow for unexpected domains, but where the dataset contained one or more of the primary, secondary and tertiary care settings, we aimed to classify as either: “hospital”
to include any inpatient or outpatient care from a secondary or tertiary care unit but without any primary or community based care data; or “primary” to include any dataset that contains, or may contain, some primary care data. We also recorded the country and medical domain of the datasets used.

Figure 1. The full workflow of papers screened for this review.

3. Results

Datasets from hospitals were used in 91% (n=130) of papers, far more than the number of papers that used, or may have used, primary care data (n=7). Additional domains that were found during data extraction were dentistry (n=4), public health (n=1) and nursing homes (n=1). Occasionally, for example with insurance datasets, it was unclear whether the dataset contained primary care data. In these instances the medical domain was used to help classify so that, for example, surgery data would be assumed to occur in hospital, but chronic conditions such as type 2 diabetes were assumed likely to contain at least some primary care data.

Of the 7 papers with a dataset which may contain primary care data: 4 used datasets from insurance providers (2 using ICD codes [9, 10], 1 using Belgian insurance codes [11] and 1 unspecified [12]), 1 had limited information about their dataset [13], 1 mentioned preliminary results but didn’t actually present them [14], and 1 had primary and secondary data for type 2 diabetes but limited results [15].

Europe (n=68) contributed the largest number of papers, though at 48% of papers it was less dominant than at the time of the previous literature review when it accounted for 73% of papers. North America (n=31) and Asia (n=22) have increased their share, while work has also appeared in South America (n=8) which was absent previously. The Netherlands (n=35) remain the country with the most papers, but second and third are now USA (n=25) and Australia (n=8); previously it was Germany and Belgium.

Oncology (n=33) is the most prevalent area, then cardiology (n=13), emergency care (n=11), stroke (n=10), surgery (n=8), diabetes (n=6), asthma (n=6) and others (n=61).

Many study challenges were identified by the authors of the reviewed papers. These include: data quality and how to assess or correct for consistency and completeness of routinely collected data; which of several competing process mining tools to use; which
of an increasing number of algorithms to consider; how to validate the results; how to give insight into the discovered processes either by improved visualizations or comprehensible models; and how to utilize the results in a clinical setting. No specific primary care challenges were mentioned in the reviewed papers, however all of these are likely to be present in primary care.

The full list of the 143 reviewed papers and the data extracted is omitted due to the restrictions on paper length. However, this information is available at https://zenodo.org/badge/latestdoi/110376986.

4. Discussion

Our review demonstrates there is little research in the area of process mining within primary care. Of the limited research, none is done exclusively in primary care.

The relevance, remit and extent of primary care varies from country to country. However, primary care plays a key role in most of the countries that we identified with at least 4 papers on process mining; only China, USA and Germany don’t require registration with a GP, or use primary care as the gatekeeper of healthcare services [16].

Acute care pathways within secondary care, where the patient is physically located within the hospital, have tight and well defined boundaries – you can monitor, interact with, and record info on the patient for the entire duration of the pathway. This is also true to some extent in outpatient settings for disease specific processes, such as cancer, when managed within specialist tertiary centres. In such cases there is a tight boundary in that all aspects of treatment are within, and recorded within, the centre. It is perhaps therefore unsurprising that these domains are popular with process mining, especially to researchers interested in method development looking for easy data sets.

Fragmentation of data may be an issue in some countries, however large primary care databases have been used for research globally with examples in USA [16], UK [17] and the Netherlands [17]. Although the boundaries of primary care are less well defined, there are still opportunities to use these data to look at processes that are exclusive to primary care. This could include various stages in chronic disease management such as monitoring, diagnosis and treatment. Medication management, and safe prescribing are other areas with potential – especially within the UK where the combination of large primary care datasets and universal electronic prescribing is particularly attractive.

The strengths of this paper are that: we have based the search on a previously published peer review, giving it increased validity; and we have explored a clinical field that is not currently well reported within the process mining community. The limitations are that: the literature search and data extraction were performed by a single author (RW) which may have introduced bias but as our intention was simply to investigate healthcare setting rather than to systematically extract more complex concepts we believe this to be sufficient; and we explicitly rejected other literature reviews that may have contained papers not found by our search, however given the breadth of our search we believe it unlikely that many papers have been missed and our results and conclusion would remain.

5. Conclusion

The lack of published papers to date suggests there are challenges to be overcome when applying process mining to primary care, so future work should look to identify and
resolve these problems. There is a wealth of primary care data available for research and a big, as yet unrealized, opportunity to analyze this data with process mining.

6. Acknowledgements

Funded by the National Institute for Health Research Greater Manchester Patient Safety Translational Research Centre (NIHR GM PSTRC). The views expressed are those of the author(s) and not necessarily those of the NHS, the NIHR or the Department of Health. NP and OJ were supported through Connected Health Cities, a Northern Health Science Alliance led programme funded by the Department of Health and delivered by a consortium of academic and NHS organizations across the north of England.

References


Publication 6: Process Mining in Primary Care: Avoiding Adverse Events Due to Hazardous Prescribing.
Process Mining in Primary Care: Avoiding Adverse Events Due to Hazardous Prescribing

Richard Williams\textsuperscript{a}, Darren M. Ashcroft\textsuperscript{b}, Benjamin Brown\textsuperscript{a}, Eric Rojas\textsuperscript{c}, Niels Peek\textsuperscript{a}, Owen Johnson\textsuperscript{c}

\textsuperscript{a}NIHR Greater Manchester Patient Safety Translational Research Centre, University of Manchester, Manchester, UK
\textsuperscript{b}Internal Medicine Department, School of Medicine, Pontificia Universidad Católica de Chile
\textsuperscript{c}NIHR Yorkshire and the Humber Patient Safety Translational Research Centre, University of Leeds, Leeds, UK

Abstract

Process mining helps understand processes within healthcare. While often used in secondary care, there is little work using primary care data. Serious adverse events that result from hazardous prescribing are common and costly. For example, non-steroidal anti-inflammatory drugs (NSAIDs) and antiplatelets can cause gastrointestinal bleeds (GiBs). Prescribing typically occurs within primary care, therefore we used this setting to attempt process mining.

Certain patients should be prescribed gastro-protection alongside NSAIDs or antiplatelets. We extracted events (drug started, drug stopped, GiB) for understanding three prescribing pathways, and applied process mining.

We found NSAIDs are often short-term prescriptions whereas antiplatelets are often long-term. This perhaps explains our finding that co-prescription of gastro-protection is more prevalent for antiplatelets than NSAIDs. We identified reasons why primary care data is harder to process mine and proposed solutions. Process mining primary care data is possible and likely useful for improving patient safety and reducing costs.

Keywords:
Process mining; Patient safety; Primary Care.

Introduction

Process mining describes a collection of methods for extracting information about processes from event logs \cite{1}. There are three distinct stages: detecting the underlying process from the event logs (process discovery); identifying deviations from what was expected (conformance checking); and generating suggestions for redesigning and improving the processes (enhancement) \cite{1}. It adds a temporal dimension to standard data mining methods. Originally applied to business processes, more recently it has been applied to other domains including healthcare. In a recent literature review, we showed that while process mining within secondary and tertiary care has become more common, there is almost no work within primary care \cite{2}.

Patient safety is fundamental to healthcare systems. Within UK primary care this is true for medication prescribing where life-threatening errors appear in 1 in 550 prescriptions \cite{3}. A recent economic analysis showed that: adverse drug reactions (ADRs) cost the NHS up to £1.6 billion a year; more than one third of ADR related hospital admissions are caused by non-steroidal anti-inflammatory drugs (NSAIDs), antiplatelets and anticoagulants; and half of the deaths associated with primary care ADRs are due to gastrointestinal bleeds (GiBs) \cite{4}. Studying the relationship between the prescribing practice of NSAIDs and antiplatelets with ADRs including GiBs is therefore important.

In the UK there are several large databases of coded primary care records available for research \cite{5}. While the quality of coding may not be universally high \cite{6}, all practice-based prescribing in primary care is electronic so therefore this would be a suitable place to attempt to apply process mining.

While the epidemiology of hazardous prescribing in primary care has been extensively studied using large electronic health record databases, to date little is known about the typical processes that lead to such prescriptions. To design effective interventions for reducing hazardous prescribing, it is essential to get a better understanding of these processes. This could lead to better decision support systems for prescribers, and ultimately improve patient safety and reduce cost by reducing the number of ADRs.

Our objective was therefore to process mine UK primary care data to explore the relationship between the prescribing of NSAIDs, antiplatelets and the adverse outcome of gastrointestinal bleeds.

Methods

A process model is a graphical representation of a process showing the events and how they interrelate via directed edges. Process discovery is the extraction of a process model from an event log via the application of an algorithm. There are many algorithms with various strengths and weaknesses. For example the \textalpha-algorithm is simple and therefore easy to understand, but it does not deal well with noisy event logs which are typical of real world processes \cite{1}. Heuristic Miner and, in particular, Fuzzy Miner are better able to deal with this noise \cite{7}. Here we focus on process discovery to prove that process mining can be applied to primary care data, and use the Fuzzy Miner to best handle the messiness of routinely collected health data.

Anonymised patient data was obtained from the Salford Integrated Record (SIR); a data warehouse with contributions from 43 general practices in Salford, UK (population 0.25M). All coded data, including diagnoses and medications, for patients who have not opted out (1.5% opt outs) was available to extract from a SQL Server database. The earliest records are historic diagnoses from the 1940s, but the bulk of the data collection is from 2000 onwards. Approval was granted by the SIR governance board and all data was obtained pseudonymised (random identifier, no name, year of birth instead of age, geographic region instead of address).

A review by Spencer et al. \cite{8} identified 56 prescribing safety indicators for use in primary care to improve patient safety. They each try to prevent a particular adverse outcome through
safer prescribing e.g. patients with chronic kidney disease should not be prescribed an NSAID because of the increased risk of acute renal failure. A subset of these indicators are included in: electronic audit and feedback initiatives such as the national PINCER [9] rollout and the SMASH intervention [10]; and clinical decision support systems such as OptimizeRx [11]. We selected three prescribing safety indicators for further analysis that focus on NSAIDs and anticoagulants, and are designed for preventing GiBs in cohorts of patients at increased risk such as the elderly and those with a history of peptic ulceration. The indicators and their descriptions used in our analyses are provided in Table 1.

Table 1– Prescribing safety indicators for preventing GiBs

<table>
<thead>
<tr>
<th>Id</th>
<th>Short name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>Age&gt;65 + NSAID</td>
<td>Patients aged 65 and over who are prescribed an NSAID should also be prescribed a gastro-protective medication (GPM).</td>
</tr>
<tr>
<td>I2</td>
<td>Pep + NSAID</td>
<td>Patients with a history of peptic ulceration who are prescribed an NSAID should also be prescribed a GPM.</td>
</tr>
<tr>
<td>I3</td>
<td>Pep + Antiplatelet</td>
<td>Patients with a history of peptic ulceration who are prescribed a antiplatelet should also be prescribed a GPM.</td>
</tr>
</tbody>
</table>

Prescription events are recorded automatically in a patient’s record, however the stopping of medication is not recorded. We have previously developed an algorithm to convert these prescription events into more meaningful events such as when a drug is started and stopped, and when a dose is changed [12]. This is done by evaluating: the date of the prescription; the amount prescribed; and the rate at which it is consumed. For the medications of interest (NSAIDs, GPMs, and antiplatelets) we extracted the start and stop events.

For each indicator we developed queries that would extract the key events for indicator I1. IQR in [square] brackets. Number of transitions in (round) brackets. NSAID – non-steroidal anti-inflammatory drug, GPM – gastro-protective medication.

<table>
<thead>
<tr>
<th>Event</th>
<th>Bleed</th>
<th>NSAID (no GPM)</th>
<th>NSAID (GPM)</th>
<th>NSAID Stopped</th>
<th>Stoped</th>
<th>GPM Started</th>
<th>GPM Stopped</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age 65</td>
<td>19,112</td>
<td>11,742</td>
<td>3,172</td>
<td>931</td>
<td>48</td>
<td>12</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>(390)</td>
<td>(1092)</td>
<td>(1385)</td>
<td>(1684)</td>
<td>(936)</td>
<td>(1617)</td>
<td></td>
</tr>
<tr>
<td>Bleed</td>
<td>14</td>
<td>22</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>(0.5)</td>
<td>(523)</td>
<td>(48)</td>
<td>(54)</td>
<td>(582)</td>
<td>(276)</td>
<td></td>
</tr>
<tr>
<td>NSAID (no GPM)</td>
<td>1.36</td>
<td>1.0</td>
<td>1.3</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(34)</td>
<td>(48)</td>
<td>(75)</td>
<td>(54)</td>
<td>(582)</td>
<td>(276)</td>
<td></td>
</tr>
<tr>
<td>NSAID (GPM)</td>
<td>1.32</td>
<td>1.3</td>
<td>0.8733</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(11)</td>
<td>(75)</td>
<td>(8733)</td>
<td>(566)</td>
<td>(566)</td>
<td>(566)</td>
<td></td>
</tr>
<tr>
<td>NSAID Stopped</td>
<td>20</td>
<td>11</td>
<td>10</td>
<td>18</td>
<td>9</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(5.57)</td>
<td>(1923)</td>
<td>(4553)</td>
<td>(563)</td>
<td>(563)</td>
<td>(563)</td>
<td></td>
</tr>
<tr>
<td>GPM Started</td>
<td>5</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(1.23)</td>
<td>(3.27)</td>
<td>(1.2)</td>
<td>(1.5)</td>
<td>(1.5)</td>
<td>(1.5)</td>
<td></td>
</tr>
<tr>
<td>GPM Stopped</td>
<td>13</td>
<td>11</td>
<td>1</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3.38)</td>
<td>(5.25)</td>
<td>(1.4)</td>
<td>(2144)</td>
<td>(2144)</td>
<td>(2144)</td>
<td></td>
</tr>
</tbody>
</table>

Results

The demographic information for the patient cohorts for each indicator are displayed in Table 2. The median duration time, interquartile range, and number of transitions between events are shown in Tables 3-5. For example, in Table 3, the event “Bleed” immediately followed the event “Age 65” in the event logs 390 times, with a median transition time of 49 (IQR [19, 112]) months. The process mining diagrams extracted from Disco are shown in Figures 1-3. The numbers on the nodes in the diagrams represent the number of times each event occurred, while the edge numbers are the number of times the target event directly followed the source event.

Table 2– Patient characteristics for each cohort and the population of Salford. Values are n (%) unless otherwise specified.

<table>
<thead>
<tr>
<th>Demographic</th>
<th>I1</th>
<th>I2, I3</th>
<th>All patients</th>
</tr>
</thead>
<tbody>
<tr>
<td># of patients</td>
<td>38,936</td>
<td>3,477</td>
<td>270,412</td>
</tr>
<tr>
<td>Age (Mean (SD))</td>
<td>76 (8)</td>
<td>66 (16)</td>
<td>37 (23)</td>
</tr>
<tr>
<td>Sex</td>
<td>Female</td>
<td>20,633 (53%)</td>
<td>1,238 (36%)</td>
</tr>
<tr>
<td>Male</td>
<td>18,303 (47%)</td>
<td>2,239 (64%)</td>
<td>138,473 (51%)</td>
</tr>
<tr>
<td>Ethnicity</td>
<td>White</td>
<td>16,291 (42%)</td>
<td>1,444 (42%)</td>
</tr>
<tr>
<td>Other</td>
<td>643 (2%)</td>
<td>139 (4%)</td>
<td>24,124 (9%)</td>
</tr>
<tr>
<td>Unknown</td>
<td>22,002 (57%)</td>
<td>1,894 (54%)</td>
<td>149,592 (55%)</td>
</tr>
<tr>
<td>Deprievation</td>
<td>1st (highest)</td>
<td>15,023 (39%)</td>
<td>1,618 (47%)</td>
</tr>
<tr>
<td>2nd</td>
<td>8,126 (21%)</td>
<td>694 (20%)</td>
<td>56,284 (21%)</td>
</tr>
<tr>
<td>3rd</td>
<td>7,536 (19%)</td>
<td>569 (16%)</td>
<td>44,770 (17%)</td>
</tr>
<tr>
<td>4th</td>
<td>3,533 (9%)</td>
<td>258 (7%)</td>
<td>19,591 (7%)</td>
</tr>
<tr>
<td>5th (lowest)</td>
<td>4,320 (11%)</td>
<td>269 (8%)</td>
<td>19,561 (7%)</td>
</tr>
</tbody>
</table>

Table 3– Median duration in months of transitions between key events for indicator I1. IQR in [square] brackets. Number of transitions in (round) brackets. NSAID – non-steroidal anti-inflammatory drug, GPM – gastro-protective medication.

For indicator I1 there were 45,479 NSAID start events. Of these, 9,981 (22%) were for patients already prescribed a GPM. A further 9,478 (21%) then started GPM at a median duration of 0 days suggesting co-prescription. However 24,106 (53%) NSAID start events were followed by an NSAID stop event at a median duration of 1 month (IQR [1, 3] months), suggesting a short term prescription without co-prescription of a GPM.
Similar results were found in indicator I2 when out of 6,368 NSAIDs, 25% (1,602) were for patients with a pre-existing GPM, 14% (861) were followed almost instantly by a GPM, and 38% (2,388) were for a short term prescription without a GPM. The results for indicator I3 suggests that GPMs are more frequently co-prescribed with APs with only 22% of AP start events (513 out of 2309) not either having a pre-existing GPM or immediately followed by a GPM.

The event most likely to precede a GI bleed or peptic ulceration is a previous GI bleed or ulceration. This is to be expected as it is known that a strong predictor of gastrointestinal adverse events is a previous bleed or ulceration.

The median duration of NSAID and GPM prescriptions is 1 month suggesting these medications are typically short-term. Antiplatelets are prescribed at a median length of 5 months and 2 months for patients with and without a pre-existing GPM respectively, suggesting longer term prescriptions.
Discussion

Summary of findings and comparison to existing literature

Little published work on process mining in primary care exists [2]; Dagliati et al. [17] used primary care data to investigate care pathways related to cardiovascular risk of Type II diabetes patients. However the majority of their data was obtained from secondary care. Another paper used primary care data, but didn’t report any results [18]. A further 4 papers used insurance data [19–22] which probably included primary care data, but also secondary care and tertiary care data. Also the level of data included in insurance datasets is different to that which is routinely collected in primary care for the provision of care. To the best of our knowledge the process mining performed for this paper is the first performed exclusively using primary care electronic health data.

GPM was more likely to be co-prescribed to patients receiving a course of antplatelets than it was to those receiving NSAIDs. The difference in prescription lengths is one possible explanation. When a clinician prescribes a short-term NSAID course, perhaps in response to an acute injury or minor illness, they may decide the risk is small enough that co-prescription of a GPM is unnecessary. However when prescribing a longer-term course the risk is increased. This might also be true for those on longer courses of NSAIDs for chronic pain conditions. Stratiﬁying medications depending on whether they are short or long term might give further insight into clinicians’ behaviour.

GPMs such as proton pump inhibitors can be prescribed for a variety of reasons. For treating an active bleed, to manage the symptoms of gastrointestinal irritation of reﬂux, or prophylactically for patients at high risk of a bleed – especially when increased because of other medications. Attempting to stratiﬁe the GPM events accordingly could again lead to more understanding of prescribing behaviour.

Implications for practice and research

In order to achieve our results there were several challenges that needed to be overcome which go some way to explain why process mining in secondary care is more prevalent.

Data quality

The quality of healthcare data is limited for many reasons. Events can be incorrectly recorded, unrecorded or encoded. All of which limit the confidence and utility of any results generated. Researchers must try and understand the limitations in their data to make best use of it. Primary care data can be thought of as snapshots of coded information that are generated on every contact with the health system. This is different for inpatient secondary care where the entire duration of treatment can be observed and recorded.

While many events recorded in a primary care systems may have uncertain veracity, the generation of a prescription is an event we can mine with conﬁdence because, in the UK, in primary care virtually all prescriptions are electronically generated. This is not true for the adverse event of bleeding which may occur in secondary care and may not be coded in the primary care record. To mitigate against this, linked primary and secondary data would be required, and is another reason why process mining exclusively in primary care is not done. Future work should focus on pathways that occur almost exclusively in primary care such as the diagnosis, monitoring and treatment of certain chronic conditions such as hypertension.

Start and end points

Within secondary care the start of a process can clearly be deﬁned as the admission to hospital, while the end of the process is discharge or death. A patient visiting hospital more than once can be treated as two separate pathways. Within primary care, processes are often cyclical and entangled with other processes. Taking indicator I2 as an example, should the start event be the ﬁrst instance of peptic ulceration, or should it be the ﬁrst prescription of an NSAID in a patient with previous peptic ulceration? The former means that each patient only has one pathway, but with potentially multiple cycles, while the latter separates each NSAID prescription into a separate process but then doesn’t take the patient’s history into account. Detailed consideration must be given to determining whether the primary care processes under analysis have well deﬁned end points, and queries structured to separate data into these individual processes.

Event granularity

The coded events within primary care are not necessarily the events that should make up the event log on which process mining depends. An example is medications where the patient record contains the prescription events, whereas the events of interest on a care pathway would be when the clinician has started, stopped, or altered the dosage of a medication. This is also true for measurements, where a series of blood pressure (BP) values do not necessarily constitute events, but the occurrence of two systolic BPs >140 mmHg within two weeks might be a trigger to investigate a diagnosis of hypertension and could therefore be considered an event to process mine. However this introduces a bias as the researcher must decide a priori what constitutes an event. Is it that a BP was taken, that the BP was high or that some combination of values were measured over a certain period of time? Careful consideration must be made to convert the raw data into an event log but this is not straightforward and is largely subjective.

Medications

The lack of a stop event for medications requires an extra processing step to determine when a patient’s medication has expired. There is also no way of knowing whether a medication once collected is in fact consumed by the patient, or if the patient is using over the counter medications. This is less of an issue in inpatient secondary care when the both the prescription and administration of medication can be monitored and recorded.

Memory

The process mining diagrams that we have produced are heuristic nets which are memory-less: each transition in the process map is taken in isolation without consideration of prior events. By redefining the start events of NSAID and antplatelets to take into account whether a GPM was already prescribed allowed us to introduce an element of memory into the system. This is useful to better understand the various pathways, however future work using other process mining modelling techniques, such as causal nets [23], might produce better results.

Conclusions

We have applied process mining to primary care clinical data on medication prescribing. There are many benefits in terms of patient outcomes and cost savings to be achieved by improving safe prescribing. Primary care data in the UK has reliably coded prescribing information and process mining can be successfully
applied leading to results that may be useful for clinical decision support systems and improving patient safety.

However primary care data presents several unique challenges. Careful pre-processing must first be undertaken, but this is subjective and must therefore be sensibly performed and meticulously recorded in order to facilitate scrutiny and reproducibility. The use of a clinical reference group to review and confirm objectively and must therefore be sensibly performed and meticulously recorded in order to facilitate scrutiny and reproducibility.

There are a range of other more powerful process mining and machine learning techniques that could be applied now that the initial problems with primary care data have been considered and to some extent addressed.

Acknowledgements

This article was funded by the National Institute for Health Research Greater Manchester Patient Safety Translational Research Centre (NIHR Greater Manchester PSTRC). The views expressed are those of the author and not necessarily those of the NHS, the NIHR or the Department of Health and Social Care.

References


Address for correspondence

Richard Williams - richard.williams@manchester.ac.uk
G303a, JMF Building, University of Manchester, M13 9PL, UK.
Publication 7: SMASH! The Salford medication safety dashboard.
SMASH! The Salford medication safety dashboard

Richard Williams
NIHR Greater Manchester Patient Safety Translational Research Centre (PSTRC), University of Manchester, Manchester, UK and MRC Health eResearch Centre, Division of Informatics, Imaging and Data Science, University of Manchester, Manchester, UK

Richard Keers
NIHR Greater Manchester Patient Safety Translational Research Centre, University of Manchester, Manchester, UK and Division of Pharmacy and Optometry, Centre for Pharmacoeconomics and Drug Safety, School of Health Sciences, Manchester Academic Health Sciences Centre (MAHSC), University of Manchester, Manchester, UK

Wouter T. Gude
Academic Medical Center, University of Amsterdam, Amsterdam, Netherlands

Mark Jeffries
NIHR Greater Manchester Patient Safety Translational Research Centre, University of Manchester, Manchester, UK and Division of Pharmacy and Optometry, Centre for Pharmacoeconomics and Drug Safety, School of Health Sciences, Manchester Academic Health Sciences Centre (MAHSC), University of Manchester, Manchester, UK

Colin Davies
NIHR Greater Manchester Patient Safety Translational Research Centre, University of Manchester, Manchester, UK and MRC Health eResearch Centre, Division of Informatics, Imaging and Data Science, University of Manchester, Manchester, UK

Benjamin Brown
NIHR Greater Manchester Patient Safety Translational Research Centre, University of Manchester, Manchester, UK and MRC Health eResearch Centre, Division of Informatics, Imaging and Data Science, University of Manchester, Manchester, UK

Evangelos Kontopantelis
NIHR Greater Manchester Patient Safety Translational Research Centre, University of Manchester, Manchester, UK and NIHR School for Primary Care Research, University of Manchester, Manchester, UK

Anthony J. Avery
NIHR Greater Manchester Patient Safety Translational Research Centre, University of Manchester, Manchester, UK and School of Medicine, University of Nottingham, Nottingham, UK

Darren M. Ashcroft
NIHR Greater Manchester Patient Safety Translational Research Centre, University of Manchester, Manchester, UK and Division of Pharmacy and Optometry, Centre for Pharmacoeconomics and Drug Safety, School of Health Sciences, Manchester Academic Health Sciences Centre (MAHSC), University of Manchester, Manchester, UK

Niels Peek
NIHR Greater Manchester Patient Safety Translational Research Centre, University of Manchester, Manchester, UK and MRC Health eResearch Centre, Division of Informatics, Imaging and Data Science, University of Manchester, Manchester, UK

ABSTRACT

Background Patient safety is vital to well-functioning health systems. A key component is safe prescribing, particularly in primary care where most medications are prescribed. Previous research has demonstrated that the number of patients exposed to potentially hazardous prescribing can be reduced by interrogating the electronic health record (EHR) database of general practices and providing feedback to general practitioners (GPs) in a pharmacist-led intervention. We aimed to develop and roll out an online dashboard application that delivers this audit and feedback intervention in a continuous fashion.
INTRODUCTION

Patient safety has become an integral part of quality management in healthcare systems worldwide. While patient safety research has traditionally focused on secondary care,\textsuperscript{1–3} primary care, as the cornerstone of modern healthcare systems,\textsuperscript{4} is increasingly recognised as an area where major improvements in patient safety can be achieved.\textsuperscript{5,6} especially due to the large numbers of medications that are prescribed on a daily basis.\textsuperscript{7} It has been shown that one in 20 prescriptions in primary care contain errors, and one in 550 contain potentially life threatening errors.\textsuperscript{7} One in 25 hospital admissions are the result of prescribing errors in primary care,\textsuperscript{8} and adverse drug reactions, of which most are avoidable, cost the NHS an estimated £500 million per year.\textsuperscript{9}

The potential for Information Technology (IT) systems to improve safety within healthcare is large and well documented.\textsuperscript{10,11} For prescribing safety, there are examples where IT systems have had positive\textsuperscript{12} and negative\textsuperscript{13,14} effects. IT-based interventions for improving prescribing safety fall broadly into two categories: clinical decision support (CDS) and electronic audit and feedback (eA&F). CDS systems, such as pop-up alerts, attempt to influence behaviour at the point of care and while some studies have shown benefits,\textsuperscript{15} others have shown that clinicians can suffer from ‘alert fatigue’ where poorly targeted alerts lead to their routine dismissal.\textsuperscript{16,17} eA&F systems, such as dashboards, provide feedback away from the point of care, usually at a population level, to allow for clinicians to review, and potentially change, their practice retrospectively.\textsuperscript{18} A systematic review of eA&F systems found a wide degree of heterogeneity in both the identified studies and their effects.\textsuperscript{19} Another literature review, specifically focusing on dashboards, had similar findings and called for more research to be undertaken to help influence the design of such systems.\textsuperscript{20} Despite the widespread usage of such dashboards, there exists little evidence as to what factors contribute to their success or failure.\textsuperscript{21}

Method Based on initial system requirements, we designed the dashboard’s user interface over three iterations with six GPs, seven pharmacists and a member of the public. Prescribing safety indicators from previous work were implemented in the dashboard. Pharmacists were trained to use the intervention and deliver it to general practices.

Results A web-based electronic dashboard was developed and linked to shared care records in Salford, UK. The completed dashboard was deployed in all but one (n = 43) general practices in the region. By November 2017, 36 pharmacists had been trained in delivering the intervention to practices. There were 135 registered users of the dashboard, with an average of 91 user sessions a week.

Conclusion We have developed and successfully rolled out of a complex, pharmacist-led dashboard intervention in Salford, UK. System usage statistics indicate broad and sustained uptake of the intervention. The use of systems that provide regularly updated audit information may be an important contributor towards medication safety in primary care.

Keywords: patient safety, drug prescriptions, electronic audit and feedback, dashboard

A common limitation of both types of intervention is that they often only indicate problems, without necessarily providing solutions. Even when specific actions are recommended, clinicians do not necessarily have the time or skills to act appropriately.\textsuperscript{22} The University of Nottingham, therefore, developed the pharmacist-led information technology intervention for reducing medication errors (PINCER) in primary care. The PINCER intervention is based on computer-generated feedback that identifies patients for whom potentially hazardous prescribing practices are present, but crucially adds educational outreach visits by trained pharmacists\textsuperscript{23} to general practices where they work with the local staff to resolve any confirmed hazardous prescribing incidents and to prevent their reoccurrence. The intervention was shown to be more effective at reducing numbers of at-risk patients than computer-generated feedback alone\textsuperscript{24}; proving the pharmacist visit plays a crucial role in effectively solving prescribing errors. It was also shown to be likely (59% chance) cost-effective in reducing prescription errors.\textsuperscript{24,25}

There are indications that the reduction in risk due to PINCER is only temporary because it does not always reduce the incident erroneous prescribing behaviour.\textsuperscript{24} This is in part because the PINCER feedback mechanism relies on snapshots of data extracted from the electronic health record (EHR) database, while feedback is known to be more effective when it is provided more than once.\textsuperscript{26} Therefore, we aimed to build upon PINCER in order to create a continuous feedback loop for cycles of quality improvement. Our objectives were to develop an application that identifies patients exposed to potentially hazardous prescribing to end users and is updated on a daily basis, and roll out the system across Salford, UK, where our previous research has shown the prevalence of potentially hazardous prescribing is greater than 5%.\textsuperscript{27}
## METHODS

### System requirements

Based on an initial scope definition created by a senior clinical pharmacy researcher (Darren M. Ashcroft) and the principal investigator of the PINCER study (Anthony J. Avery), it was decided that a front-end dashboard application and the associated back end was required that would:

1) receive, validate and process data extracts of patient records from general practitioner (GP) systems on a daily basis, via an existing shared care record infrastructure – the Salford Integrated Record (SIR);
2) execute queries to the GP system data, based on pre-defined indicators, to identify patients at risk of adverse medication events;
3) present lists of at-risk patients to GPs and pharmacists, in a secure environment, restricting the visibility of patient identifying information to those clinicians that are responsible for that patient’s care;
4) provide the results in a timely, user friendly and actionable manner.

The indicators to be used in the dashboard were selected from the set of 56 prescribing safety indicators for GPs identified by Spencer et al. Indicators were selected based on their severity and the practicality of extracting the relevant information from clinical records. They aim to prevent: gastrointestinal bleeding; asthma exacerbations; acute kidney injury; liver damage and neutropenia; hypo and hyperthyroidism and thrombotic risks. The list of indicators in the dashboard is available in Table 1 in the Supplementary Material.

Computable representations of the indicators were required which would involve creating clinical code sets for each component, and determining whether a patient was prescribed to take a medication at a given point in time. This last requirement is not simple as although prescription events in UK primary care are always recorded electronically, there is usually no coded record when a medication is stopped, so instead the termination of the medication must be inferred by the lack of a repeat prescription.

It was also decided that the system’s feedback regarding each indicator would consist of a numerator (also known as the affected or ‘at-risk’ patients), and a denominator (also known as the eligible patients). The ‘affected patients’ are those who have breached the conditions of the indicator and are therefore those patients at risk from potentially hazardous prescribing who need corrective action. The ‘eligible patients’ are those who meet a particular subset of the conditions of the indicator and are the population of patients against which the affected patients can be measured. The terms numerator and denominator are used as the proportion of eligible patients who are also affected patients can be used to compare between practices of different sizes and demographics.

### Engineering methods

The initial design of the dashboard’s user interface was created following an iterative process which was informed by short interviews with key stakeholders involving six GPs, seven clinical commissioning group (CCG) pharmacists and one member of our patient and public involvement group. Prototype dashboard designs were reviewed by the stakeholders during interview and feedback sought. This feedback was incorporated into the design for the next dashboard iteration.

The software architecture of the dashboard, the user interface and all the associated back-end processes were designed by the lead author Richard Williams. The software development was undertaken by Richard Williams and Colin Davies. Where possible open-source technologies were used to ensure that the system could be made publicly available in the future if required. Operating systems and server architecture were designed in order to meet the requirements of the secure server hosting within the Salford Royal Foundation Trust (SRFT) data centre.

### System usage monitoring

A final requirement was that the system should track usage, and record all interactions with the dashboard down to individual mouse clicks, hovers and non-sensitive key strokes, that is, not passwords. A large volume of usage tracking data will therefore be collected enabling us to report on the frequency and duration with which the dashboard is used, who its primary users are (e.g. pharmacists or GPs), and how this varies between practices and over time. It will also allow us to assess which feedback modalities provided by the dashboard (table, benchmark charts, trend charts and patient lists) are typically accessed by users, and under which circumstances.

Finally, we can analyse which areas of medication safety (i.e. which indicators) users tend to focus on when they access the dashboard.

### Implementation and roll-out

Any general practice in Salford was eligible for receiving the intervention provided they had access to SIR. Research team members met with Salford CCG leads and GP quality leads to identify practices for recruitment. Once identified, the CCG pharmacy team (supported by the research team) provided practice managers/senior partners with an information sheet about the intervention for distribution amongst practice staff and an invitation to take part.

If practices decided to take part, their staff received online access to the dashboard. No physical installation at the practice level is necessary. In addition, trained NHS clinical pharmacists will visit the practices regularly for at least 12 weeks, access the dashboard, and advise on changes in medication prescribing/monitoring in collaboration with practice staff.

Pharmacist time was resourced by Salford CCG and by the NIHR Greater Manchester Patient Safety Translational Research Centre (http://www.patientsafety.manchester.ac.uk/). Pharmacists were recruited and trained to help deliver the intervention. The training was based on the original PINCER trial pharmacist training and involved: introducing the prescribing safety indicators included in the dashboard and the risk they pose to patients; the importance of academic...
detailing and causation analysis/systems thinking; describ-
ing the SMASH dashboard and clinical pharmacist interven-
tion and developing the understanding of how to respond to
to potentially hazardous prescribing in practice. Each pharma-
cist was assigned to one or more practices with whom they
arranged an initial meeting during which they talked to the
practice about the indicators and showed them how to use
the dashboard. Pharmacists and GPs were encouraged to
focus on resolving cases of potentially hazardous prescribing
identified by the dashboard during an initial 12-week ‘inten-
sive’ period. From the date of the initial meeting with each
general practice, we followed up that practice for 12 months
to discover: whether the numbers of at risk patients reduce;
whether any reduction is sustained; how people use the sys-
tem and how their use affects patient outcomes.

RESULTS

Architecture
The system is deployed to two servers (see Figure 1), running
Windows Server 2012, in the secure data centre of the SRFT.
One server runs the main patient database using SQL Server
2012, and the other runs SMASH – the web application that
the users access. SMASH is rendered on the client side with
AngularJs, and is served with data via a RESTful API run-
ing on NodeJS and Express. It is available on the secure
NHS N3 network meaning only people on that network can
access the dashboard. Firewalls ensure that only recruited
practices can view the dashboard. Users can access the site
remotely using a virtual private network (VPN) connection to
their practice.

Each day at midnight, the system receives data from pri-
mary care EHR data sources. Currently, this is SIR, a data
warehouse containing all primary care data from Salford,
though the system allows any suitable primary care data
source to be incorporated. Once the data is updated, another
batch runs to execute SQL stored procedures against
the patient data to generate lists of at-risk patients for
the front-end web application. These report data are copied to
MongoDB on the web application server. The entire pro-
cess takes about 2 hours.

Indicator development
We have developed computable representations of each
indicator. Clinical code sets were created for each com-
ponent of the indicator. For example, the indicator that
looks for asthmatic patients who are prescribed a non-
selective beta blocker required two main code sets: one
code set to identify asthmatic patients and one to identify
beta blocker prescriptions. In fact, a third set of codes for
the event ‘asthma resolved’ was required so that patients
whose asthma is no longer active would not be detected
by the indicator.

To determine when a patient was prescribed to take a med-
ication, we used two approaches. The first method was to
count any patient with a prescription in the previous \( n \) months,
where in our case, and for the current PINCER indicators, \( n = 3 \) months, but with longer periods for medications, such as combined hormonal contraceptives (\( n = 6 \) months), which often have longer prescriptions. The second approach was to reuse an algorithm that we had previously developed that considers the date of the prescription, the amount prescribed and the rate of medication usage (‘take 2 a day’) to determine when the patient has likely stopped the medication without the occurrence of another prescription.35

**System functionality**

For users assigned to a single practice, the first screen after logging in presents a summary table for their practice on today’s date. For users with multiple practices, the user must first select a practice before viewing the summary. A second tab presents detailed information for each indicator (Figure 2 – all data in this and future screenshots is fictitious). The number of affected and eligible patients, the proportion of eligible patients affected and the current CCG average can be viewed here. A comparison date, defaulting to 30 days ago, allows the system to show the user: how many affected patients have been resolved; how many remain affected and how many new cases there have been since this comparison date. The user can: change the date of the report and the comparison date; sort the table by each column; select certain indicators to always appear on top irrespective of the sorting and drill down into lists of patients by clicking on any number within the table. A third tab displays this information in charts. One example visually compares the user’s practice with the average across the CCG (Figure 3). Another example, not displayed here, shows how the practice’s performance changes over time.

Selecting one of the hyperlinked numbers within the table in Figure 2 takes the user to a screen with a list of patients affected by the selected indicator on the selected date (Figure 4). For each patient, we display: their NHS number, which indicators they are breaching, and for how long they have been affected. The clinician can look up the patient in the practice’s EHR to determine an appropriate course of action. The selected indicator and date of interest can be changed.

From the patient list (Figure 4), the user can select a tab named ‘Information’ to display a summary of the evidence supporting the selected indicator (Figure 5). This includes: the risk to patients; links to the academic literature supplying the evidence; the consequences of inaction and possible actions that could be taken.

Finally, some users are given the role of ‘CCG user’ which allows them to see summary performance measures across all practices within the CCG. They can view the proportion of

![Figure 2 System screenshot: indicator list for a practice. It shows number and proportion of patients affected, compares performance with the past, and allows the user to view lists of patients by clicking the hyperlinked numbers. CKD = chronic kidney disease; NSAID = nonsteroidal anti-inflammatory drug; GastProt = gastrointestinal protective medication; Warf = warfarin; NOAC = novel oral anticoagulant; LABA = long-acting beta-adrenoceptor agonist; ICS = inhaled corticosteroid](image)
at-risk patients in each practice for a single indicator, or for all indicators, in table or chart form (Figure 6).

Pharmacist and practice recruitment
By October 2017, 36 pharmacists, three pharmacy technicians and two CCG managers were trained to use the dashboard and how to introduce it to general practices. Although pharmacists and general practices were encouraged to focus on an initial 12-week ‘intensive’ period, in many cases, pharmacists continued to monitor and use the dashboard after this time.

We recruited 43 out of 44 general practices within Salford. The missing practice wanted to be involved but does not contribute to SIR and so was ineligible for the intervention. The first practice recruited completed the 12-month follow-up on 17 April 2017, and the last practice recruited will complete in September 2018. Effects on prescribing safety and adverse events will be evaluated thereafter in an interrupted time series analysis.

Usage
As of 17 November 2017, there are 135 registered users (51 pharmacists, 48 GPs and 36 other practice staff such as nurses and technicians) in the dashboard, excluding developers and research team members, with 28 users (all pharmacists) logging into the system in the last month (since 18 October 2017). In 2017, there have been an average of 91 user sessions per week (SD = 39), with an average duration of 17 minutes.

Preliminary findings
On 31 January 2017, we conducted an interim analysis comparing recruited practices with at least 1 month’s usage, with those practices not yet recruited or with less than 1 month’s usage. The number of at-risk patients in recruited practices \( (n = 32) \) had fallen from 1444 on 1 January 2016 to 882 on 31 January 2017. This is a mean reduction of 17.6 patients per practice and is significant \( (p = 0.0002) \) when compared with the mean reduction of 2.1 patients in practices not yet recruited \( (n = 12) \).

DISCUSSION
We have developed SMASH, a dashboard for displaying patients ‘at-risk’ of a serious adverse event, such as acute...
Figure 4 System screenshot: a list of patients currently flagged by SMASH for the ‘asthma and beta blocker’ indicator. It shows the NHS number, which indicators the patient is breaching, and when they first flagged for this indicator.
Patients with a history of asthma who have been prescribed a ß blocker

What is the risk to patients?

In susceptible patients ß blockers can precipitate acute attacks of bronchospasm or worsen daily symptoms resulting in mortality or low grade morbidity respectively. The BNF advises that "ß blockers should be avoided in patients with a history of asthma or bronchospasm; if there is no alternative, a cardioselective ß blockers can be used with extreme caution under specialist supervision. Atenolol, bisoprolol, metoprolol, nebivolol, and (to a lesser extent) acenbutolol, have less effect on the ß1 (bronchial) receptors and are, therefore, relatively cardioselective, but they are not cardioprotective. They have a lesser effect on airways resistance but are not free of this side effect." The Committee on Safety of Medicines issued the following advice: "...ß blockers, even those with apparent cardioselectivity, should not be used in patients with asthma or a history of obstructive airways disease, unless no alternative treatment is available. In such cases the risk of inducing bronchospasm should be appreciated and appropriate precautions taken."

What evidence is there that this pattern of prescribing is harmful?

ß blockers vary in their affinity for ß1- and ß2-adrenoceptors, and are divided into two groups, cardioselective (affinity for ß1), and non-cardioselective (affinity for ß2). The majority show little selectivity for one receptor over the other, except for bisoprolol (14-fold greater affinity for ß2-adrenoceptors) and timolol, sotalol and propranolol (20-fold, 12-fold, and 8-fold greater affinity for ß2-adrenoceptors, respectively).

Table 1: Cardioselective and non-cardioselective ß blockers

<table>
<thead>
<tr>
<th>Cardioselective ß blockers (relative selectivity for ß1-adrenoceptors)</th>
<th>Non Cardioselective ß blockers (relative selectivity for ß2-adrenoceptors)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acenbutolol (2.4)</td>
<td>Labetalol (2.5)</td>
</tr>
<tr>
<td>Nebivolol (7.3)</td>
<td>Betaxolol (1.7)</td>
</tr>
</tbody>
</table>

**Figure 5** System screenshot: detailed information and evidence about the asthma and beta blocker indicator

**Figure 6** System screenshot: chart showing the proportion of eligible patients who are affected for each practice within the CCG
have since stopped their medication. It also has the problem that there may be nothing actionable the GP or pharmacist can do to remove the patient from the dashboard – they must instead wait until their most recent medication is before the window of inspection. This has the potential to discourage usage of the dashboard and could also lead to extra work as users repeatedly review a patient’s record. The algorithmic approach is more promising, but requires additional pre-processing work to map drug codes to ingredients and strengths.

One limitation is that we have built a system external to the GPs standard EHR systems and so users are required to view patient lists within SMASH, before looking the patients up in a separate system. A more integrated approach would be easier for the end user, though given the high levels of usage that we have observed, it is a problem that the users are perhaps willing to overlook. Future versions of GP systems should consider incorporating safety indicators.

Over the coming years, we will: improve the existing dashboard based on feedback from the trial; deploy more indicators; roll out the system across Greater Manchester and working with industry partners and explore ways of allowing patients to interact with the system. The role of patients in their safe care is important and the ability for patients to discover when they are ‘flagged up’ by safety systems such as this will start to change the interactions between the patient and provider, and opens up several interesting avenues for future research.

Acknowledgements

This research was funded by the National Institute for Health Research Greater Manchester Patient Safety Translational Research Centre (NIHR Greater Manchester PSTRC). The views expressed are those of the author(s) and not necessarily those of the NHS, the NIHR or the Department of Health and Social Care. MRC Health eResearch Centre grant MR/K006665/1 supported the time and facilities of EK. Connected Health Cities is a Northern Health Science Alliance led programme funded by the Department of Health and delivered by a consortium of academic and NHS organisations across the north of England. The work uses data provided by patients and collected by the NHS as part of their care and support. The views expressed are those of the author(s) and not necessarily those of the NSHA, NHS or the Department of Health.

REFERENCES


Table 1 Descriptions of the 13 indicators deployed in the current dashboard and the clinical aim of each

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
<th>Clinical aim</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Prescription of an oral NSAID without co-prescription of an ulcer-healing drug in a patient aged ≥65 years.</td>
<td>Prevent gastrointestinal bleeding.</td>
</tr>
<tr>
<td>2</td>
<td>Prescription of an oral NSAID without co-prescription of an ulcer-healing drug to a patient with a history of peptic ulceration.</td>
<td>Prevent gastrointestinal bleeding.</td>
</tr>
<tr>
<td>3</td>
<td>Prescription of an antplatelet drug without co-prescription of an ulcer-healing drug to a patient with a history of peptic ulceration.</td>
<td>Prevent gastrointestinal bleeding.</td>
</tr>
<tr>
<td>4</td>
<td>Prescription of warfarin or NOAC in combination with an oral NSAID.</td>
<td>Prevent gastrointestinal bleeding.</td>
</tr>
<tr>
<td>5</td>
<td>Prescription of warfarin or NOAC in combination with an antplatelet drug without co-prescription of an ulcer-healing drug.</td>
<td>Prevent gastrointestinal bleeding.</td>
</tr>
<tr>
<td>7</td>
<td>Prescription of a non-selective beta-blocker to a patient with asthma.</td>
<td>Prevent acute asthma exacerbation.</td>
</tr>
<tr>
<td>8</td>
<td>Prescription of a long-acting beta-2 agonist inhaler (excluding combination products with inhaled corticosteroid) to a patient with asthma who is not also prescribed an inhaled corticosteroid.</td>
<td>Prevent acute asthma exacerbation.</td>
</tr>
<tr>
<td>9</td>
<td>Prescription of an oral NSAID to a patient with heart failure.</td>
<td>Prevent thrombotic risk.</td>
</tr>
<tr>
<td>10</td>
<td>Prescription of an oral NSAID to a patient with chronic renal failure (eGFR &lt; 45) already prescribed an ACE inhibitor/ARB and loop diuretic</td>
<td>Prevent acute kidney injury.</td>
</tr>
<tr>
<td>11</td>
<td>Prescription of an oral NSAID to a patient with chronic renal failure (eGFR &lt; 45)</td>
<td>Prevent acute kidney injury.</td>
</tr>
<tr>
<td>12</td>
<td>Prescription of Methotrexate without both a recent full blood count and a recent liver function test.</td>
<td>Prevent liver damage (liver function test) and the development of neutropenia (full blood count).</td>
</tr>
<tr>
<td>13</td>
<td>Prescription of Amiodarone without a thyroid function test.</td>
<td>Prevent hypo and hyperthyroidism.</td>
</tr>
</tbody>
</table>

NOAC = novel anticoagulants such as apixaban, dabigatran or rivaroxaban. Antplatelet drugs are aspirin, clopidogrel, prasugrel and ticagrelor. Peptic ulceration includes upper gastrointestinal bleeds, but does not include peptic ulcer surgery, gastritis, duodenitis or oesophageal varices. Ulcer-healing drugs include the PPIs and H2-antagonist – it does not include Misoprostol, Sucralfate or Bismuth.
Epilogue

Clinical codes have been a large part of my working life over the last few years. I have searched for them, I have created code sets for use in several e-A&F tools and for defining cohorts in research studies, I have researched how others use and manage them, and I have created software for the construction and validation of code sets. In that time I have also become aware of the quirks, problems and inconsistencies that crop up in most clinical code terminologies. I was fortunate to win a Science Slam at MIE2018 when I gave a presentation highlighting some of the more bizarre clinical codes (T011300 – Train collision with fallen tree, pedal cyclist injured), and also retelling several classic fairy tales using only clinical codes (TG3y500 – Accident caused by spinning machine). I followed this up by extending the presentation into an article, which was published in the prestigious Christmas edition of the BMJ. This paper, although lacking a certain scientific rigour, does raise some important points regarding clinical code terminologies. It is included here for light relief.
A Christmas guide to clinical coding

With the rollout of a new coding system in the UK under way, Richard Williams reveals a lighter side to recording clinical data

Richard Williams senior software engineer and research fellow
NIHR Greater Manchester Patient Safety Translational Research Centre, University of Manchester, UK

♫ Sleigh bells ring, are you listening?
Read codes fade, SNOMED’s glistening ♫

If you didn’t find the section heading amusing you either have no sense of humour (SNOMED: 288801003—“Unable to use humour”), you don’t know any Christmas songs (SNOMED: 16170002—“Music blindness”), or, like most normal people, you don’t know much about clinical coding.

In a nutshell, coding is the use of short alphanumeric codes to record symptoms, diagnoses, laboratory tests, procedures, and medicines in the electronic health record. For example, a GP in UK primary care might enter “C10F” to record a diagnosis of type 2 diabetes, “22A” along with a value and a unit to record a person’s weight, “di1m” to record a prescription of 300 mg soluble aspirin tablets, or “T550” for spacecraft launch pad accidents (more on this later). Typically, a clinician starts typing and the medical record software suggests appropriate codes from a dropdown list.

Collectively, these codes are called either dictionaries (because each code has a definition) or terminologies. C10F, 229, and di1m are all examples of Read codes, which were created in the 1980s by James Read for use in primary care. Read codes have been the main clinical coding system in UK primary care since the mid-1990s.

♫ And since we’ve no more Read code, let it SNO, let it SNO, let it SNO(MED) ♫

SNOMED is another terminology, one that is internationally recognised and used in more than 50 countries. Unlike Read, SNOMED codes are entirely numeric. The UK government recently decided that the NHS would migrate all clinical systems to use SNOMED by 2020, with primary care the first to migrate.

The full primary care rollout was expected to happen by autumn 2018 but, given that it’s been over 15 years since the move to SNOMED was first suggested, it’s unsurprising that it’s behind schedule. Having just one terminology, instead of many, will hopefully increase the accuracy and reliability of the data exchanged between care settings, to enable better reporting and analysis.

Should auld code sets be forgot and never brought to mind?
Certainly not. In this season of reflection it seems topical and timely to look back at Read code and look ahead to the transition to the new system—not only because Read code has played such a big part in primary care over the past two decades but also because “SNOMED” sounds a lot like “snowmen”.

♫ Codes for falling, all around me, GPs coding, having fun ♫

The Read code dictionary is immense. I’m sure that all GPs remember the first time they treated a child who had gone to...
school only to fall off a cliff and thought, “There’s a code for that!” (U10F200: fall from cliff; occurrence at school) or the first time they had to deal with farm based ice skating injuries (U102700: fall involving ice skates, skis, roller skates, or skateboards; occurrence on farm). Falls are a big problem, and Read codes offer various options beyond the common occurrence of frail elderly people falling in their own homes, such as falling from a turret (TC25.00), a haystack (TC4y100), or a flagpole (TC23.00).

You can fall “in an aircraft” (T532.00) or “from an aircraft” (T534.00). I know which one I’d prefer. But you can also fall “on an aircraft” (T533.00), which I assume means falling from something else (also coded, perhaps) and then landing on a plane. Finally, for absolute completeness, there’s a catch-all code in the form of T53z.00: “Fall in, on, or from aircraft not otherwise specified.”

Read codes are a dream come true for a stuntman’s doctor. You can fall from a cable car (T613.00) or from a moving vehicle (T183.00) into a dock, pit, or storm drain (TC3y100, TC3y300, TC3Z100)—and, more spectacularly, from a burning apartment, a burning camping place, a burning barn, a burning church, a burning factory, a burning hospital, a burning school, or a burning mobile home (TD07).

Almost every incident represented by a code has probably happened to someone somewhere but, without wishing to diminish their suffering or indeed death, is it clinically useful to know whether people who have drowned did so when a boat overturned (T400.00), sank (T401.00), or was crushed (T404.00)—or that they fell (T402.00) or jumped (T403.00) from said boat because it was on fire?

**Superfluous terms**

The odd “Easter egg” is harmless, but there are 523 Read codes for falls, 1480 for motor vehicle traffic incidents, 459 for motor vehicle NON-traffic incidents, and 1104 for air, sea, and train incidents. Together these represent a significant proportion of the entire dictionary, and their inclusion is a small but cumulatively significant drain on clinicians’ time while searching for codes that are actually useful.

Don’t worry if you’re a GP who regularly uses these codes. As SNOMED drew extensively on Read codes during its creation, they’re all still available in the new dictionary. Fortunately, SNOMED codes (unlike Read codes) can be marked as inactive to remove superfluous terms. One hopes that SNOMED’s creators pursue this actively and ruthlessly.

♫ It’s beginning to look a lot like . . . quiz time ♫

It wouldn’t be Christmas without a pantomime, so see if you can guess the following patients from the meticulously coded casebook of the Doctors Grimm (figs 1-2):

---

**Fig 1** An apple a day keeps the doctor away, usually
Fig 2 It's Christmas—go on, let your hair down

Or how about this tale, from a general practice somewhere in the Middle East (fig 3):

Fig 3 Patient in a stable condition
Finally, SNOMED, with its large number of non-clinical codes (including the kitchen sink: 4909001), is even better than Read for storytelling. Here’s the tale of a day in the life of a clinically obese deliveryman (fig 4):

♫ Look to the future now, it’s only just begun ♫

It’s a laudable aim to have a single code dictionary across all parts of the NHS, and this may well lead to the promised improvements. It may, however, have unintended consequences: at almost five times the size of Read, SNOMED has the potential to lengthen the time GPs take to code patient interactions.

Everyone involved in healthcare stands to gain from well coded data. The quality of coding in primary care increased after the implementation of the Quality and Outcomes Framework.¹ This would indicate barriers to good coding, and it seems to have taken an obvious and quantifiable incentive (in this case, financial) to improve matters. I suspect that most GPs would prefer simply to type brief notes during a consultation than to undertake the laborious and error prone task of identifying the correct clinical codes. But why not have the best of both worlds? Natural language processing—the ability for computers to process text intelligently—is not ready to replace manual coding, but it’s already sufficiently advanced to analyse a written note and suggest codes that accurately reflect the encounter.² And why stop there? Speech recognition, combined with natural language processing, could generate a transcript of a conversation between a GP and a patient so that, by the time GPs turn to their computers, they would have a summary of the encounter plus possible diagnoses, a range of appropriate medicines, and suggested next steps. (“The robot will see you now…”)³

Delivering the goods

After the migration to SNOMED, without improvements to existing IT systems the extra moments spent searching through an even longer list of irrelevant codes generated by each search will increase the burden on clinicians, who are already under tremendous time pressure. It’s essential that barriers to good coding are identified and removed, whether through training, IT improvements, or wholesale reform of the code dictionary.⁴ Research is urgently needed on the design of such systems, their usability in practice, and whether SNOMED is in fact delivering the goods in a real world setting.

What, then, does the future hold for clinical code terminologies, GPs, and the NHS in general? God only knows (Read: R2yz.11; SNOMED: 301327002).

This article was funded by the National Institute for Health Research Greater Manchester Patient Safety Translational Research Centre. The views expressed are the author’s and not necessarily those of the NHS, NIHR, or the Department of Health and Social Care.

Competing interests: I have read and understood BMJ policy on declaration of interests and declare the following interests: none.

Provenance and peer review: not commissioned; not externally peer reviewed.

1 Benson T. The history of the Read Codes: the inaugural James Read Memorial Lecture 2011. Inform Prim Care 2011;19:173-82.22688227
5 For an interesting discussion of this issue see: Spence D. Unreadable codes. BMJ 2010;341:c6694.10.1136/bmj.c6694.