BMJ Health & Care Informatics

Development of a customised programme to standardise comorbidity diagnosis codes in a large-scale database

Robert C Osorio ,[1] Kunal P Raygor,[2] Adib A Abla[2]

To cite: Osorio RC, Raygor KP, Abla AA. Development of a customised programme to standardise comorbidity diagnosis codes in a large-scale database. *BMJ Health Care Inform* 2022;**29**:e100532. doi:10.1136/bmjhci-2021-100532

Received 15 December 2021
Accepted 09 April 2022

[1]School of Medicine, University of California San Francisco, San Francisco, California, USA
[2]Department of Neurological Surgery, University of California San Francisco, San Francisco, California, USA

Correspondence to
Robert C Osorio;
robert.osorio@ucsf.edu

ABSTRACT

Objectives The transition from ICD-9 to ICD-10 coding creates a data standardisation challenge for large-scale longitudinal research. We sought to develop a programme that automated this standardisation process.

Methods A programme was developed to standardise ICD-9 and ICD-10 terminology into one system. Code was improved to reduce runtime, and two iterations were tested on a joint ICD-9/ICD-10 database of 15.8 million patients.

Results Both programmes successfully standardised diagnostic terminology in the database. While the original programme updated 100 000 cells in 12.5 hours, the improved programme translated 3.1 million cells in 38 min.

Discussion While both programmes successfully translated ICD-related data into a standardised format, the original programme suffered from excessive runtimes. Code improvement with hash tables and parallelisation exponentially reduced these runtimes.

Conclusion Databases with ICD-9 and ICD-10 codes require terminology standardisation for analysis. By sharing our programme's implementation, we hope to assist other researchers in standardising their own databases.

INTRODUCTION

On 1 October 2015, the department of Health and Human Services updated the International Classification of Diseases (ICD) system by mandating the adoption of ICD-10 diagnosis codes in electronic medical records.[1] Serving as the new standard for naming and categorizing patient diagnoses, the ICD-10 system contains over five times more codes than ICD-9, posing a challenge for analysing longitudinal databases spanning both systems. Prior solutions have included the use of alternate coding systems, which are updated each time a new ICD system is released. Current literature is aimed at the accuracy and scope of these systems,[2 3] how they update with new ICD releases,[3 4] and how systems are similar or different.[5 6] These studies fail to address how to implement such a system on a large-scale database, where manual reference and cell-by-cell translation is infeasible. We sought to develop a programme that quickly and accurately standardises a dataset to one diagnostic coding system.

METHODS

A nationwide dataset of paediatric hospital discharges was examined. Originating from the Healthcare Cost and Utilisation Project (H-CUP), this Kids' Inpatient Database (KID) contained administrative data on 15.8 million hospital discharges across 2003–2016. The targets of our data manipulation were 20 columns of diagnosis codes that represented patient comorbidities at the time of surgery: while most cases in the database occurred during ICD-9's era, 3.1 million discharges (19.6%) occurred in the 2016 KID update, and thus had ICD-10 codes. As a solution to this difference, H-CUP offers Elixhauser Comorbidity Software, which assigns diagnosis names to comorbidities based on the ICD-9 or 10 system.[7] Prior to programme development and testing, we defined a successful programme as one which cross-referenced all ICD-10 codes to their corresponding comorbidity classification. The resulting database would contain all 15.8 million discharges using the same classification system.

Prior to development, a Microsoft Excel File was acquired from H-CUP, which listed ICD-10 diagnosis codes in the first column, and Elixhauser diagnosis names in the first row. The remaining cells were marked with a '1' if an ICD-10 code matched a corresponding comorbidity. This served as the 'dictionary' for our data translation. All computer code was developed and executed on RStudio, V.4.0.2.

A programme was written to examine each column of comorbidity data and extract any ICD-10 diagnosis codes encountered. Each code was individually compared with the 'dictionary': the programme scanned through rows until it found a matching ICD code, then scanned across that row until a '1' was seen (denoting it found a matching

Table 1 Computer program development, runtimes and relative efficiency

Program name	Time to complete 100 000 rows	Time to complete entire 3.1M translations	Relative efficiency
Linear programme	12.5 hours	16.1 days	1×
Parallelised programme with hash table	1.2 min	38 min	610.1×

A programme was developed that successfully standardised the comorbidity coding system used in a 15.7 million patient database spanning 2003–2016. Parallelising this programme and implementing a hash table increased the speed by more than 600-fold, allowing 3.1 million patient rows to be updated in under 40 min.

diagnosis). When a match was found, the column name (the diagnosis) was captured, and the corresponding column in the KID was marked as a '1,' denoting that patient as having this comorbidity. This process repeated until all diagnosis codes were translated in that patient row. The programme would then proceed to the next row in the database, and would start over on the new ICD-10 codes.

During development, code was tested on a random 1000 rows of data. Once it successfully translated these rows, the programme was deployed on the 3.1 million patients with ICD-10 codes. A duplicate of the programme was then created, and served as the starting point for runtime optimisation. In a similar fashion to the development of the original code, this new programme was tested on a random 1000 rows, then executed on the larger database.

RESULTS

Both programmes successfully translated ICD-10 codes to the Elixhauser comorbidity classification. Results on programme runtimes for the first iteration ('Linear') and the more efficient ('Parallelised') code are displayed in table 1. When testing runtimes for the linear code, it updated 100 000 rows in 12–13 hours, varying slightly in each test. As a result, this linear code would take 16 days to complete the 3.1 million target rows in our dataset. Programme testing was stopped after 7 days due to impracticality of runtime.

In development of a second iteration of code, runtime was reduced by targeting algorithm efficiency. Complexity was improved through conversion of the 'dictionary' into a hash table, exponentially reducing the number of computer operations performed. Runtime was further improved by breaking the data into subsets, and translating each subset simultaneously. On a computer with a 16-core processor, this allowed the 3.1 million discharges to be broken into 16 subsets of roughly 200 000 discharges. This parallelised code translated all 3.1 million rows in 38 min (1.2 min/100 000 samples), a more than 600-fold increase in processing speed compared with the original programme.

DISCUSSION

For longitudinal databases spanning across the 2010s, researchers face the challenge of analysing data that utilises both ICD-9 and ICD-10 codes. Prior literature addressed the creation and accuracy of standardised classification systems, but failed to discuss how to implement these systems on large databases where manual translation is impossible.[2–6] We successfully automated the standardisation of diagnostic terminology for a database of 15.8 million hospital discharges across 2003–2016. Databases of this size often pose a challenge for automated programmes, as evidenced by our initial programme's excessively long runtime. The subsequent programme we developed, however, ran more than 600 times faster, underscoring the significance of code quality in large scale data manipulation.

The largest gains in runtime can be attributed to the implementation of hash tables instead) of a 'dictionary' Excel file. When a computer iterates through an Excel dictionary of R rows and C columns, up to R * C comparisons are needed to find a match for just one comorbidity. When translating up to 20 comorbidities per row, for 3.1 million datapoints, these accumulate to roughly 62 million * R * C computer operations, guaranteeing excessive runtimes. A hash table is a data structure composed of a list of 'keys,' where each key is associated with one and only one 'value'. By converting our dictionary into a hash table with ICD-10 diagnosis codes as 'keys' and Elixhauser's comorbidity names as 'values,' translating diagnoses became exponentially simpler. Whereas the dictionary required R*C operations to find a match for a single ICD-10 code, a hash table requires just one action by the computer.

In addition to reducing programme complexity, code parallelisation also contributed to its faster runtimes. By splitting the data into 16 subsets to simultaneously translate, our programme ran 16 times faster. This parallelisation is possible due to multicore processors available in computers sold today.

Other advantages in the development of a customised programme include generalisability to future implementations. Our programme examines the number of processing cores on the computer running the algorithm, ensuring that data are always divided and analysed as efficiently as possible. Additionally, our programme should be easily implemented on any 'dictionary' that is plugged into our software, so that future systems such as ICD-11 may also be translated. Any 'dictionary' of reference values may be used, ensuring long-term utility of our algorithm in future practice of large-scale research.

CONCLUSION

Hash tables and parallelised code allowed us to standardise the coding system used by a 15.8 million patient

database in under 40 min. We hope that by publishing our methods of translation on such a notably large database, we aid researchers in transforming other large datasets. When attempting to standardise data spanning multiple years, researchers should consider programming such as ours where hash tables and parallelisation allow extreme amounts of data review to be completed in an exponentially quicker time frame.

Contributors Conception and design: AA, RCO and KR. Acquisition of data: AA, RCO and KR. Analysis and interpretation of data: AA, RCO and KR. Drafting the article: RCO and KR. Critically revising the article: AA, RCO and KR. Reviewed submitted version of manuscript: AA, RCO and KR. Approved the final submission of the manuscript: AA. Overall study supervision: AA.

Funding The authors have not declared a specific grant for this research from any funding agency in the public, commercial or not-for-profit sectors.

Competing interests None declared.

Patient consent for publication Not applicable.

Provenance and peer review Not commissioned; externally peer reviewed.

Data availability statement Data may be obtained from a third party and are not publicly available. Data are available through the Healthcare Cost and Utilisation Project.

ORCID iD

Robert C Osorio http://orcid.org/0000-0002-7669-2176

REFERENCES

1 The switch from ICD-9 to ICD-10: when and why. Available: https://icd.codes/articles/icd9-to-icd10-explained [Accessed 23 Oct 2021].
2 Feudtner C, Feinstein JA, Zhong W, *et al*. Pediatric complex chronic conditions classification system version 2: updated for ICD-10 and complex medical technology dependence and transplantation. *BMC Pediatr* 2014;14:199.
3 Glasheen WP, Cordier T, Gumpina R, *et al*. Charlson Comorbidity Index: *ICD-9* Update and *ICD-10* Translation. *Am Health Drug Benefits* 2019;12:188-197.
4 Glasheen WP, Renda A, Dong Y. Diabetes Complications Severity Index (DCSI)-Update and ICD-10 translation. *J Diabetes Complications* 2017;31:1007–13.
5 Hua-Gen Li M, Hutchinson A, Tacey M, *et al*. Reliability of comorbidity scores derived from administrative data in the tertiary hospital intensive care setting: a cross-sectional study. *BMJ Health Care Inform* 2019;26:e000016.
6 Brusselaers N, Lagergren J. The Charlson comorbidity index in registry-based research. *Methods Inf Med* 2017;56:401–6.
7 Elixhauser comorbidity software, version 3.7. Available: https://www.hcup-us.ahrq.gov/toolssoftware/comorbidity/comorbidity.jsp [Accessed 27 Oct 2021].