A prototype software framework for transparent, reusable and updatable health economic models

Matthew P Hamilton[1,2,*] Caroline X Gao[2,3,1] Glen Wiesner[4] Kate M Filia[2,3] Jana M Menssink[2,3] Petra Plencnerova[5] David Baker[2,3] Patrick D McGorry[2,3] Alexandra Parker[6] Jonathan Karnon[7] Sue M Cotton[2,3] Cathrine Mihalopoulos[1]

Abstract

Summary: Computers are tools that are now essential to the work of health economists. However, the ethical dimensions of how health economic models are implemented computationally are poorly understood. Explicit ethical standards for computational health economic models (CHEMs) amd the technical infrastructure to facilitate implementations of that meet those standards are required. We propose six criteria for assessing ethical implementation of CHEMs – two each for the three domains of transparency, reusability and updatability. To facilitate the implementation of CHEMs that meet these criteria, we developed a novel prototype software framework in the open source programming language R. The framework comprises six code libraries that collectively provide a toolkit for authoring CHEMs, supplying them with data and using them to undertake generalisable and transferable analyses. The framework supports integrations with existing digital services for collaborative software development and data archiving. We are currently applying the software framework to develop and apply utility mapping models in youth mental health. We assess the first set of utility mapping CHEMs that we have developed with the framework as wholly meeting both transparency criteria (open access code and data and clarity about author contributions and beliefs), one reusability criteria (liberal terms of use) and one updatability criteria (infrastructure for model maintenance) and partially meeting the two remaining criteria for transparency (supports generalisability and transferability) and updatability (retesting and deprecation). The assessment criteria and the software framework we have developed can inform future work to understand and improve ethical computational implementations of health economic models.

Code: Visit https://www.ready4-dev.com for more information about how to find, install and apply the prototype software framework.

[1] School of Public Health and Preventive Medicine, Monash University, Clayton, Australia
[2] Orygen, Parkville, Australia
[3] Centre for Youth Mental Health; The University of Melbourne, Parkville, Australia
[4] Heart Foundation, Melbourne, Australia
[5] headspace National Youth Mental Health Foundation, Melbourne, Australia
[6] Victoria University, Footscray, Australia
[7] Flinders University, Adelaide, Australia

[*] Correspondence: Matthew P Hamilton <matthew.hamilton1@monash.edu>

1 Introduction

Health economics is a discipline concerned with problems that arise due to scarce resources, such as how to value health and healthcare, allocate healthcare budgets and configure health services [1]. In seeking to solve these problems, health economists typically use models which are simplified and selective representations of systems that are believed to influence human health. These representations can be described in words and pictures (a conceptual model), in equations (a mathematical model) or in computer code (a computational model). The predictions reported in health economic studies are typically generated by the execution of a computer program which applies a computational model to compatible data inputs (e.g., parameter values) and performs a sequence of numeric calculations.

Computational models are now widely used to inform health policy and system design [2–5]. This level of influence has concomitant ethical responsibilities for model developers that are often poorly understood and inadequately fulfilled [6–9].

Computational health economic models (CHEMs) can be implemented using specialized commercial software or authored as bespoke software projects in a programming language. Advantage of commercial modelling tools are simplicity and ease of use, but a software development approach facilitates development of models that may be more realistic, transparent, reusable and adaptable [10]. For CHEMs that are implemented as software projects, a major early decision is selection of an appropriate software framework. A software framework is a shared common technology used by developers to collaboratively author software and which is not typically visible to software end-users [11]. Advantages of using software frameworks include facilitating code reuse and extension, promoting good programming practice and the capability to provide enhanced functionality and performance without additional effort by developers [12]. However, software frameworks can be challenging and time consuming to create [12] and then difficult for others to learn, often requiring developers to undergo specialist training [11]. There is also a risk that a software framework may become excessively complex over time [12].

We are developing a model to explore multiple economic questions relating to the mental health of young people aged 12 to 25. When making choices about how to implement this model computationally, we wished to facilitate ethical development and use. However, we are not aware of any software framework for implementing CHEMs that adhere to explicitly stated ethical requirements.

In this paper, we describe: (i) a set of ethical responsibilities for CHEM developers and criteria for assessing responsible CHEM implementations; (ii) a prototype software framework for the ethical implementation of CHEMs; and (iii) use of the software framework to develop a computational economic model in youth mental health, with an initial focus on outcome valuation.

1.1 Ethical requirements and assessment criteria

We considered prior literature on modelling practice, our own professional experience and the needs of our project, to identify three core ethical responsibilities of CHEM developers, three attributes of CHEMs that enable fulfilment of these responsibilities and six criteria (two for each model attribute) for assessing ethical CHEM implementations.

1.2 Ethical responsibilities of CHEM developers

We believe that health economists have ethical responsibilities relating to the social acceptability, adequacy for purpose and beneficial impact of their computational models.

Misalignment between the values of model developers and those of the population groups affected by decisions based on their models presents significant ethical risks [8,13]. The value judgments of model developers influence the assumptions, selection of model features and standards for evidence that shape the CHEM development process [14]. These value judgments are rarely made explicit, omissions that may lead to socially unacceptable policy recommendations [13]. For example, to reduce the risk of inequitable policy implementation, it may be important for of a model to predict the benefits of harms of an intervention for different sub-populations [6], but model developers may prefer to allocate a project budget to implementing other model features [8].

Health economists have duties both to take sufficient care that a CHEM is adequate for the explicit purpose for which it was developed and to provide potential third party users with the means of assessing its adequacy for their proposed purposes [3,6,15,16]. However, it is common for CHEMs to have serious methodological flaws [17,18]; insufficient validation [19–21], poor reproducibility [22–24]; and undeclared errors [25].

Even an acceptable and adequate CHEM will have limited beneficial impact if it not much used, if it is mis-used or when its acceptability and adequacy rapidly decay. The scientific goals of generalisability (application without adaptation) and transferability (selective reuse and/or modification of model components) [26] are advanced when models are appropriately reused in new contexts. Reuse of models as components of other models can also make model implementation more efficient [27,28].

Common barriers to model re-use include commercial and legal considerations [15,29], as well as challenges related to model transferability across jurisdictions [28]. The temporal window for valid application of CHEMs is often limited by implementation choices that rarely facilitate routine updates [30]. Without ongoing maintenance, a model may become less reliable with time [28], deterioration that model users may be unaware of, and has a growing risk of being deployed for purposes for which it is poorly suited [9].

1.3 Criteria for assessing ethical CHEM implementation

The acceptability, adequacy and public benefit responsibilities for CHEM developers are easier to state than to measure. It may therefore be pragmatic for assessment criteria for ethical modelling practice to instead be based on measurable attributes of the models themselves. As described in Table 1, we believe that implementing CHEMs that are transparent, reusable and updatable (TRU) can enable modellers to meet their ethical obligations. We therefore selected these model attributes to use as the basis for deriving ethical assessment criteria.

Transparency has been recommended as a core criterion for assessing ethical public health modelling practice [6]. Guidance on transparency in health economic modelling recommended that model code and data should be clearly documented, potentially with different versions for technical and non-technical users [31]. Notably, the same guidelines, published over ten years ago, did not include recommendations on sharing model code and data. However, more recent guidance recommends publicly dissemination of healthcare model artefacts using online repository services [3]. Repositories such as Zenodo [32] and Dataverse [33] provide persistent storage solutions that generate a Digital Object Identifier (DOI) for each code and data collection. An essential component of quality assuring health economic models is verification - ensuring that calculations are correct and consistent with model specifications [34]. The extensiveness of verification checks in models implemented as software projects can be reported using the concept of code coverage [35] - the proportion of model code that has been explicitly tested. Tests should ideally combine both unit tests (to verify that small, isolated sections of code produce the correct output when run independently) and acceptance tests (to verify that the correct output is produced when multiple code components are run together to perform tasks that meet core user-requirements [36]). The nature and extent of individual model authorship contributions can become unclear when models are implemented over longer time-frames with a large and changing group of collaborators [8]. This issue can be addressed by use of online repository services such as GitHub [37], that provide citation tools and can transparently record all individual code contributions to a modelling project over its lifecycle.

Assessment criteria for a transparent CHEM:

- T1: All model code, non-confidential data and testing procedures and outcomes are available in open access repositories.
- T2: It is easy to see who developed and tested each part of the model and to identify the modelling team's assumptions, judgments and theories about model development and use.

Making a CHEM's code, data and documentation publicly available is helpful but insufficient for promoting model re-use. The choices that CHEM developers make about model implementation and licensing will also shape who can use a model and for what purposes. Using open-source development platforms and licenses can aid both generalizability and transferability. Compared to using commercial modelling software,

Table 1: Framework standards

Objective	Standard	Meaning
Accountable - it is easy to see who developed, tested and applied a model and how they did it.		
	A1	Open online repositories are used to permanently archive, uniquely identify and transparently record authorship and development history of model code and data.
	A2	Model code and data are documented.
	A3	Model code uses consistent and abstracted syntax.
	A4	Literate programming is used to implement model analyses.
	A5	Model code coverage is reported.
	A6	All parts of a study analysis and reporting workflow can be reproduced and/or replicated.
	A7	Model code and data are distributed with tools to support appropriate citation.
Reusable - a model and its components can be used in other models and by other modellers.		
	R1	Model code is made available for re-use under copyleft or permissive licenses.
	R2	Non-confidential model data is licensed for liberal re-use (potentially subject to ethical use terms).
	R3	Model code and data are stored and managed separately.
	R4	Model code uses encapsulation and inheritence for data structures.
	R5	Model code uses functions to implement algorithms.
	R6	Model code is distributed as code libraries.
	R7	Test data is available to demonstrate the transferability of model code.
	R8	Statistical models are distributed with tools for making out of sample predictions.
	R9	User-interfaces allow non-technical users to configure and use models.
Updatable - a model and its components are maintained and continuously improved.		
	U1	Model code and data are version controlled.
	U2	The signficiance and status of code and data updates are indicated with semantic versioning and release types.
	U3	Continuous integration is used to verify model code updates.
	U4	Deprecation conventions are used to retire model code and data.

authoring CHEMs in an open-source language like R [38] makes it easier to store model algorithms and data in distinct files and locations (as opposed to hard coding - embedding data such as parameter values into source code) which facilitates selective modification of model components. This benefit can be further enhanced if model developers adopt a modular approach, in which a model is constructed from multiple reusable and replaceable sub-models (modules) [39]. To grant permissions to others to use, test and adapt models and their components, health economists can avail of two broad categories of open source licensing options. Some guidance strongly recommends the use of permissive licensing [40] that provides users with great flexibility as to the purposes (including commercial) for which content can be re-used. An alternative approach is to use copyleft licenses [41] that can require content users to distribute any derivative works they create under similar open-source arrangements.

Assessment criteria for a reusable CHEM:

- R1: Model code and data are implemented to facilitate both generalizability and transferability.
- R2: Terms of use allow anyone to reuse model code and non-confidential data, in whole or in part, without charge, and for purposes that include the creation of derivative works.

To remain valid for longer, models should be continually updated and refined as new evidence emerges and healthcare systems evolve [42,@ 28]. Ensuring that a model is regular reviewed to identify and implement required improvements is a receommended defence against model validity decay [9]. Key enablers of sustainable maintenance of open source research software are committed, adequately resourced core development team and active user community [43]. Currently, the core development team for a CHEM will be typically be funded to produce a project end-point deliverable whose specifications are well defined early in the project. For more complex and multi-purpose CHEMs, particularly those designed to be incorporated into decision support systems, it may be better for development teams to adopt Agile Software Development [44], an approach that has been recommended for complex public health software projects [45]. An Agile model will be less clearly specified in the initial project plan, but will instead continually develop in response to the requirements and feedback of users, who are provided with an initial, simplified working version of the model at the earliest feasible opportunity. Online communities can be an efficient means of engaging model users in testing each version of a model, identifying issues and suggesting improvements. Services such as GitHub [37] provide collaborative code development tools [46] that help elicit, integrate and reconcile contributions from multiple contributors and to ensure each update is uniquely identifiable and retrievable. It is important that verification checks are rerun with each model update, a task that can be automated using the software development practice of Continuous Integration [47]. The risk of model revisions having unintended consequences for third party users can be mitigated through the use of deprecation conventions [48] that take an informative and staged approach to retiring outdated model code and data.

Assessment criteria for an updatable CHEM:

- U1: Resources and infrastructure are in place to support sustained development, testing, maintenance and version control of a model in collaboration with model users.
- U2: Each new release of a model is retested, with changes implemented to minimize disruptions for existing model users.

1.4 Software framework

To support the collaborative development of CHEMs that meet TRU assessment criteria, we have created a prototype software framework called ready4. We have designed this framework as a lightweight extension to existing software development infrastructure that could be used to implement CHEMs that are both open-source (for transparency and reusability) and modular (for reusability and updatability).

1.5 Framework libraries

To work within the popular open-source programming environment R [38], the ready4 software framework is implemented as six development version R code libraries. The R libraries collectively provide model developers with tools to combine R's functional and object-oriented programming paradigms [49] in authoring CHEMs, supplying those CHEMs with data and using CHEMs to implement reproducible modelling analyses. The six novel libraries and the preexisting third-party R libraries they depend on are summarised in Table 2.

Standardization and interoperability are core requirements of implementing a modular approach. Model modules need to be able to share inputs and outputs with each other and to be run as independent models [50]. Each CHEM module authored with the ready4 framework will include both a data structure (specifying the required properties of data that can validly be supplied to a module) and a set of algorithms (specifying the operations that can be performed on data contained in a module instance). The foundational framework library, called ready4, defines a template module data structure (using R's S4 class system) from which all CHEM module data structures will be created and a novel syntax that enable module algorithms to be consistently named. The ready4 library also contains tools for retrieving web based information on model modules, datasets and analysis programs authored with the framework and for partially automating updates to a project documentation website.

Three R libraries are designed to help standardize workflows for authoring, documenting, testing and disseminating new model modules. The ready4pack library is designed to integrate with GitHub and provides tools for authoring model modules and desseminating them as themed bundles in R libraries libraries that are:

- documented (with a website, a PDF manual itemising selected contents and a PDF manual itemising all contents);
- licensed (using the copyleft GNU GPL-3 [66] by default);
- easily citable (citation information can be retrieved within an R session or from hosting repositories); and
- quality assured (each update triggers continuous integration workflows, including any acceptance and unit tests created by module library authors).

The ready4pack library depends on two other module authoring libraries. Writing model algorithms as collections of functions (short, self-contained and reusable software routines that each perform a discrete task), has been recommended as good practice for scientific computing [40]. The ready4fun library contains tools for authoring functions in a consistent house style that automatically generates basic documentation for each function. Functions to implement model algorithms can be associated with a module via a special type of function called a method. Tools from the ready4class library can help streamline and standardise the authoring of module data structures and their associated methods and to automatically generate basic documentation for each module.

The ready4use library contains tools for supplying model modules with data stored in online repositories (hosted on a Dataverse installation or on GitHub), labelling these datasets and then sharing them via online repositories. The ready4show library contains tools to help author R Markdown programs that combine model modules and datasets to undertake analyses. These programs are either self-documenting (code is easy to understand and integrated with plain English explanations of what it does) or trigger the creation of separate documents (e.g. a scientific manuscript).

1.6 Framework integration with online services

Our software framework needs to be used in conjunction with a number of online services. To facilitate its application to our youth mental health project, we established and configured accounts with these required services (see Availability of data and materials).

Table 2: Modelling toolkit R libraries for developing and using MOSHEMs that meet framework standards

Package	Focus	Standard	Depends on these R libraries
ready4	Foundation	A3 R4	assertthat bib2df dataverse dplyr fs Hmisc kableExtra knitr lifecycle magrittr methods natmanager piggyback purrr readr readxl rlang rmarkdown rvest stats stringi stringr testit testthat tibble tidyRSS tools utils zen4R
ready4fun	Module algorithms	A2-3 R5	desc devtools dplyr generics gert Hmisc knitr lifecycle lubridate magrittr methods piggyback pkgdown purrr readxl ready4 ready4show ready4use rlang sinew stats stringi stringr testit testthat tibble tidyr tools usethis utils xfun
ready4class	Module structures	A2 R4-5	devtools dplyr fs gtools Hmisc knitr lifecycle magrittr methods purrr ready4 ready4fun ready4show rlang stats stringi stringr testit testthat tibble tidyr usethis utils
ready4pack	Module libraries	A1-2,5,7 R1,3,6 U1-3	dataverse dplyr knitr lifecycle magrittr methods purrr ready4 ready4class ready4fun rlang stringr testthat tibble tidyr utils
ready4use	Datasets	A1-2,7 R3,7 U1	data.table dataverse dplyr fs Hmisc knitr lifecycle magrittr methods piggyback purrr readxl ready4 ready4show rlang stats stringi stringr testit testthat tibble tidyr utils
ready4show	Analyses	A4,6	dataverse DescTools dplyr flextable grDevices here Hmisc kableExtra knitr knitrBootstrap lifecycle magrittr methods officer purrr ready4 rlang rmarkdown stringi stringr testthat tibble tidyr utils xtable

We created a GitHub organisation (a collection of code repositories) where all code (libraries, programs and sub-routines) that we author with the framework is stored and version controlled. We configured individual repositories in our GitHub organisation that are used for code library projects to use GitHub actions to implement continuous integration. By default, code libraries authored with our framework will use continuous integration to assess compliance with policies specified by the Comprehensive R Archive Network (CRAN) [51]. To track our code coverage, we linked our GitHub organisation to an account we established at codecov [52]. To facilitate the creation and hosting of documentation websites, we enabled GitHub Pages in each repository we used for code library development. We also created a Zenodo community - a collection of permanent, uniquely identified repositories. We then linked our Zenodo community and GitHub organisation so that every time we specify a version of code in one of our GitHub repositories as a "release", a copy of that code is automatically created on Zenodo with a DOI. Finally, to manage model datasets, we created a dedicated collection within the Harvard Dataverse installation.

2 Application

2.1 Economic topics

Currently, we are using the ready4 software framework to develop, apply and share youth mental health computational models in four of the twelve domains of health economics identified by Wagstaff and Culyer [1]:

- health and its value (our projects: utility mapping models);
- determinants of health and ill-health (our projects: models for creating synthetic household populations with key risk and protective factors for mental disorders);
- demand for health and health care (our projects: spatial epidemiology and help-seeking choice models); and
- supply of health services (our projects: a model of primary mental health care services).

Once these projects are completed, our aim is to flexibly combine these models as modules of a multi-purpose model (also called "ready4") to answer questions in two additional Wagstaff and Culyer domains:

- efficiency and equity (our goal: assess the distributional impacts and identify the optimal targeting of care provision); and
- economic evaluation (our goal: assess the cost-utility of competing policy options for improving the mental health of young people).

2.2 Case study: Utility mapping

Our initial application of the ready4 software framework was to undertake a previously described study [53] to develop utility mapping models for use in samples of young people presenting to primary mental health services. The ready4 software framework was used to develop CHEM modules, supply those modules with data and implement modelling analyses, creating the following artefacts:

- development version module libraries for describing and validating youth mental health human record datasets [54], scoring health utility [55], specifying utility mapping models [56] and implementing reproducible utility mapping studies [57];
- a development version library of functions for finding and using utility mapping models developed with these tools [58];

- data collections of synthetic populations for testing model modules [59] and study input and results data [60];
- programs for replicating all steps from data ingest to manuscript reporting [61], applying utility mapping models to new data [62] and generating a synthetic representation of the study dataset [63];
- subroutines for creating a catalogue of utility mapping models [64] and generating a draft scientific manuscript [65] for studies implemented with these modules.

We created a checklist (Table 3) that we used to subjectively assess these study outputs against TRU criteria. For each criterion, we provided a global assessment of whether it was met using the responses "yes", "no" or "partial". We believe the outputs from our utility mapping study may be assessable as having satisfactorily met four of the six criteria (T1, T2, R2 and U1) and to have partially met two criteria (R1 and U2). The main shortcomings that we identified when applying the assessment criteria were the need for additional development before the CHEM modules would be sufficiently generalizable for valid application in datasets that measure health utility with different instruments and a general lack of unit testing.

2.3 Model documentation

We developed a versioned model documentation website (www.ready4-dev.com) that provides guidance to model developers on how to use and contribute improvements to the ready4 software framework and model. The documentation website was developed using the Hugo framework [66], Docsy theme [67] and Algolia DocSearch [68] and is hosted using the Netlify [69] service. We used functions from the ready4 R library to partially automate website updates relating to available CHEM modules, datasets and analysis programs. We linked our Netlify account to our GitHub organisation so that the project website would automatically update whenever the source code in its GitHub repository was edited.

Table 3: Framework checklist applied to utility mapping study

	Standard	Met?	Description
A1	Public	Yes	Datasets are available at https://doi.org/10.7910/DVN/DKDIB0 and https://doi.org/10.7910/DVN/HJXYKQ. For details of where to find Development and Archive code see: https://www.ready4-dev.com/docs/getting-started/software/libraries/types/module/ https://www.ready4-dev.com/docs/getting-started/software/executables/programs/ https://www.ready4-dev.com/docs/getting-started/software/executables/subroutines/
A2	Documentation	Yes	All code libraries have documenting websites with URLs that concatenate 'https://ready4-dev.github.io/' and the library name (e.g. https://ready4-dev.github.io/youthvars). All three Markdown programs are self-documenting, with one [61] including additional instructions in a README file. Only one sub-routine [65] is documented with a meaningful README file. All datasets have meaningful metadata descriptors.
A3	Syntax	Yes	All libraries are authored with ready4pack [70] to ensure a consistent house style. All libraries except [58] use framework syntax, as does one program [61].
A4	Literate programming	Yes	All programs use literate programming.
A5	Code coverage	No	No current reporting of code coverage.
A6	Reproducibility	Yes	All parts of the study workflow from raw data ingest through to data processing, analysis, reporting and dissemination of study outputs can be reproduced (if granted access to source data) or replicated (using supplied synthetic data) with one program [61].
A7	Citation tools	Yes	All libraries can return citation details when running R's 'citation' function. All code and data repositories have tools to generate citation details.
R1	Code licenses	Yes	All code libraries, programs and sub-routines use GPL-3 licenses.
R2	Data terms	Yes	Datasets use amended version of template provided by Harvard Dataverse [71].
R3	Separation	Yes	All development code is stored on repos in a GitHub organisation [72] and all archived code releases are available in a Zenodo community [73]. All non-confidential data are stored in repositories within a Harvard Dataverse collection [74].
R4	Encapsulated, inheriting	Yes	Four [54–57] out of five libraries include encapsulating and inheriting modules. See: https://www.ready4-dev.com/docs/model/finding-modules/
R5	Functions	Yes	All code libraries include functions. The most complete list of functions for each library is available by clicking the 'Manual - Developer (PDF)' link on each package's documentation homepage (see item A2 above). The relationship between functions can be illustrated using an app: https://www.ready4-dev.com/docs/getting-started/software/libraries/dependencies/
R6	Libraries	Yes	All module data-structures and algorithms are distributed as code libraries [54–58].

	Standard	Met?	Description
R7	Test data	Yes	Two synthetic datasets and their data dictionaries are publicly available in a data repository [59]. One (ymh_clinical_tb.RDS) closely resembles the study dataset and was released so that the main study algorithm [61] can be rerun by those without access to the confidential study dataset. The other (eq5d_ds_tb.Rds) is deliberately different to the source dataset in both variable naming convention and the concepts used for predictors and outcome measures. It was created to demonstrate the transferability of study algorithms.
R8	Prediction tools	Yes	Model catalogues (PDF files beginning with 'AAA_TTU_MDL_CTG') are available in the study results dataset [60] and describe the predictive performance of all models under a variety of usage regimes (including when the source dataset in the R model object is replaced with fake data). The youthu library [58] includes tools for searching for and applying models compatible with different types of input data. An example program to demonstrate this functionality is available in both RMarkdown [62] and rendered PDF formats (the 'Application.pdf' file in the study results dataset [60]).
R9	User interface	No	No user interface has yet been developed.
U1	Version controlled	Yes	All code is version controlled using git [75] and GitHub [37]. All source code is available in a GitHub organisation [72].
U2	Semantic versioning	Yes	Semantic versioning is used in all code. As no code library has yet been submitted to CRAN [51], only the development version extensions of each version number have been incremented to date.
U3	Continuously integrated	Yes	All five libraries use continuous integration (CI). CI results for each library can be viewed at a URL that concatenates 'https://github.com/ready4-dev/', the library name and '/actions' (e.g. https://github.com/ready4-dev/youthvars/actions)
U4	Deprecation	Yes	Retired code is deprecated using tools from the lifecycle R library [76] (e.g. everything after "## DEPRECATED FNS" in https://github.com/ready4-dev/youthvars/blob/main/data-raw/fns/add.R). Package vignettes and datasets are also deprecated e.g. https://ready4-dev.github.io/youthvars/articles/Replication_DS.html)

3 Discussion

Ethical practice is a core expectation of health researchers and computational methods underpin most quantitive research, yet an understanding of what constitutes ethical computational modelling practice in health is underdeveloped [6]. The modeller responsibilities, enabling model attributes and model implementation assessment criteria that we propose can help address this gap.

The ethical responsibilities and enabling model attributes we describe have both commonalities and distinctive features compared to a previous ethical framework for computational modelling in public health [6]. The authors of that framework propose 13 questions to evaluate ethical risk across the four criteria of independence, transparency, benficience and justice. Their descriptions of the four criteria at least partially map to either our proposed modeler responsibilities ("justice" to "social acceptability", "independence" to "adequacy for purpose" and "beneficience" to "beneficial impact") or enabling model attributes ("transparency"). However, while our six assessment criteria are specific to three attributes (TRU) of the computational implementation of the model, the prior ethical framework includes questions relevant to the the conceptual and mathematical models and the potential impacts of model use. Examples of these more general evaluation questions include (for the justice criterion) "is any lack of knowledge about important parameters attributable to uncertainty or variability?" and (for the beneficience criterion) "if a policy is based on the model evidence, is it more likely to be effective and beneficial than a decision made in the absence of the model?". The less numerous and more focused assessment criteria we propose may potentially be more tractable to implement in reviews of models authored by third parties and as the basis for designing software frameworks to support ethical computational model implementation.

Currently, many if not most existing CHEMs are insufficiently transparent [19,22–24], reusable [15,77] and updatable [30,45]]. There appears to be in-principle support from health economists to address these practice shortfalls through greater use of for open-source approaches [29] that are currently rarely implemented [15,77]. Existing incentive structures for health economists generally do not promote facilitating peers to reuse their work. Currently, it takes "an extraordinary amount of idealism" to dedicate the substantial time and resources required to author, test, document and maintain even fragile prototype research software that could instead be used to write scientific manuscripts [78].

Reducing waste in research is a core responsibility of research funders [79] and funding the development of CHEMs that are not adequately understood, reused or updated is wasteful. Previously recommended strategies for more beneficial health economic research investments include support for harmonized ethical standards for model development [6], methodological innovation to improve model transferability [80], networks of modellers working on common health conditions [81], and centralized infrastructure such as open source model repositories [29] and a standard platform for model implementations [19]. Development of software frameworks to support ethical CHEM implementations could enable and enhance each of these strategies.

As illustrated by Table 3, we have developed a software framework that can help us to author a youth mental health model that largely satisfy our TRU criteria. However, we believe our software framework is currently too fragile to be anything more than a prototype for supporting the development needs other modelling teams and projects. A major reason for this distinction is that our software framework was developed with the needs of only one group of developers in mind – ourselves. We currently lack the resources required to adequately implement strategies to target factors such as user enjoyment, usability, active user-community and supporting resources that influence adoption of software frameworks [11].

Our prototype framework has a number of features that subsequent work to develop ethical software frameworks may find useful to incorporate. Firstly, developing a software framework to work within an existing and widely used open source programming language such as R or python, can keep framework scope relatively narrow (making it more tractable to develop, maintain and learn) while readily leveraging and coherently integrate other modelling tools written in that language (e.g. the dependency libraries we list in Table 2). Secondly, implementation that combines both object oriented and functional programming paradigms can avail of the modular and syntactical simplicity benefits of the former, while limiting needless bundling of code artefacts (a limitation of object oriented approaches famously described as: "you wanted a banana but what you got was a gorilla holding the banana" [82]). Thirdly, a sensible trade-off needs to be found between transparent code implementation (which requires clear and sufficiently detailed documentation) and Agile Software Development (for which a foundational principle is prioritizing the development of working code over writing documentation [44]). Our software framework makes this trade off by enforcing the use of consistent code naming conventions and file organisation which in turn enables automated generation of simple documentation at every code update. All model data-structures and algorithms are therefore always documented (at least minimally, with machine authored content), meaning model developers have a requirement to write customized documentation less frequently.

A future software framework for ethical CHEMs would ideally incorporate a base set of features useful to developers of computational models across all domains of public health, with the capability for community-led extensions that are tailored to the needs of modellers focused on specific health-conditions.

4 Conclusion

We have identified criteria that can be used to systematically assess extent to which the computational implementation of health economic models adheres to the ethical goals of transparency, reusability and updatability. We have developed an open-source software framework that can support the ethical computational implementation of economic models in youth mental health. Our framework can be used as a prototype for developing future software frameworks to support ethical implementation of CHEMs.

Acknowledgement

The authors would like to acknowledge the contribution of John Gillam who provided advisory input to this research.

Availability of data and materials

The most up to date and comprehensive source of documentation on our framework and model is available at https://www.ready4-dev.com . Development versions of all code repositories referenced in this article are available in https://github.com/ready4-dev/ . Archived code releases are available in https://zenodo.org/communities/ready4 . All data repositories referenced in this article are available in https://dataverse.harvard.edu/dataverse/ready4 .

Ethics approval

Software framework development did not involve human subject research and was not ethically reviewed. The utility mapping worked example is a previously reported study that was reviewed and granted approval by the University of Melbourne's Human Research Ethics Committee, and the local Human Ethics and Advisory Group (1645367.1).

Funding

Software framework development was funded by Orygen, VicHealth, Victoria University and an Australian Government Research Training Program (RTP) Scholarship . The utility mapping study used as a worked example was funded by the National Health and Medical Research Council (NHMRC, APP1076940), Orygen and headspace.

Conflict of Interest

None declared.

References

1. Wagstaff A, Culyer AJ. Four decades of health economics through a bibliometric lens. Journal of health economics. Elsevier; 2012;31: 406–439.
2. Dakin H, Devlin N, Feng Y, Rice N, O'Neill P, Parkin D. The influence of cost-effectiveness and other factors on nice decisions. Health economics. Wiley Online Library; 2015;24: 1256–1271.
3. Erdemir A, Mulugeta L, Ku JP, Drach A, Horner M, Morrison TM, et al. Credible practice of modeling and simulation in healthcare: Ten rules from a multidisciplinary perspective. Journal of translational medicine. 2020;18: 369. doi:10.1186/s12967-020-02540-4
4. Lo NC, Andrejko K, Shukla P, Baker T, Sawin VI, Norris SL, et al. Contribution and quality of mathematical modeling evidence in world health organization guidelines: A systematic review. Epidemics. 2022;39: 100570. doi:https://doi.org/10.1016/j.epidem.2022.100570
5. Christen P, Conteh L. How are mathematical models and results from mathematical models of vaccine-preventable diseases used, or not, by global health organisations? BMJ Global Health. BMJ Specialist Journals; 2021;6. doi:10.1136/bmjgh-2021-006827
6. Boden LA, McKendrick IJ. Model-based policymaking: A framework to promote ethical "good practice" in mathematical modeling for public health policymaking. Frontiers in Public Health. 2017;5. doi:10.3389/fpubh.2017.00068
7. Pliakos EE. Ethics in economic modeling in health care. AMA Journal of Ethics. American Medical Association; 2021;23: 599–600.
8. Thompson E. Escape from model land: How mathematical models can lead us astray and what we can do about it. New Yourk: Basic Books; 2022.
9. Calder M, Craig C, Culley D, De Cani R, Donnelly CA, Douglas R, et al. Computational modelling for decision-making: Where, why, what, who and how. Royal Society open science. The Royal Society Publishing; 2018;5: 172096.
10. Incerti D, Thom H, Baio G, Jansen JP. R you still using excel? The advantages of modern software tools for health technology assessment. Value in Health. Elsevier; 2019;22: 575–579.
11. Myllärniemi V, Kujala S, Raatikainen M, Sevońn P. Development as a journey: Factors supporting the adoption and use of software frameworks. Journal of software engineering research and development. SpringerOpen; 2018;6: 1–22.
12. Edwin NM. Software frameworks, architectural and design patterns. Journal of Software Engineering and Applications. Scientific Research Publishing; 2014;2014.
13. Duckett S. A journey towards a theology of health economics and healthcare funding. Theology. SAGE Publications Sage UK: London, England; 2022;125: 326–334.
14. Harvard S, Werker GR, Silva DS. Social, ethical, and other value judgments in health economics modelling. Social Science & Medicine. 2020;253: 112975. doi:https://doi.org/10.1016/j.socscimed.2020.112975
15. Feenstra T, Corro-Ramos I, Hamerlijnck D, Voorn G van, Ghabri S. Four aspects affecting health economic decision models and their validation. PharmacoEconomics. 2022;40: 241–248. doi:10.1007/s40273-021-01110-w
16. Thompson EL, Smith LA. Escape from model-land. Economics. De Gruyter Open Access; 2019;13.
17. Carletto A, Zanuzzi M, Sammarco A, Russo P. Quality of health economic evaluations submitted to the italian medicines agency: Current state and future actions. International Journal of Technology Assessment in Health Care. Cambridge University Press; 2020;36: 560–568. doi:10.1017/S0266462320000641
18. Wonder M, Dunlop S. Assessment of the quality of the clinical evidence in submissions to the australian pharmaceutical benefits advisory committee: Fit for purpose? Value in Health. 2015;18: 467–476. doi:https://doi.org/10.1016/j.jval.2015.02.011

19. Ghabri S, Stevenson M, Möller J, Caro JJ. Trusting the results of model-based economic analyses: Is there a pragmatic validation solution? Pharmacoeconomics. 2019;37: 1–6. doi:10.1007/s40273-018-0711-9
20. Kolovos S, Bosmans JE, Riper H, Chevreul K, Coupé VM, Tulder MW van. Model-based economic evaluation of treatments for depression: A systematic literature review. PharmacoEconomics-open. Springer; 2017;1: 149–165.
21. Haji Ali Afzali H, Gray J, Karnon J. Model performance evaluation (validation and calibration) in model-based studies of therapeutic interventions for cardiovascular diseases: A review and suggested reporting framework. Applied health economics and health policy. Springer; 2013;11: 85–93.
22. Jalali MS, DiGennaro C, Guitar A, Lew K, Rahmandad H. Evolution and reproducibility of simulation modeling in epidemiology and health policy over half a century. Epidemiologic Reviews. 2021;43: 166–175. doi:10.1093/epirev/mxab006
23. McManus E, Turner D, Sach T. Can you repeat that? Exploring the definition of a successful model replication in health economics. Pharmacoeconomics. 2019;37: 1371–1381. doi:10.1007/s40273-019-00836-y
24. Bermejo I, Tappenden P, Youn J-H. Replicating health economic models: Firm foundations or a house of cards? PharmacoEconomics. 2017;35: 1113–1121. doi:10.1007/s40273-017-0553-x
25. Radeva D, Hopkin G, Mossialos E, Borrill J, Osipenko L, Naci H. Assessment of technical errors and validation processes in economic models submitted by the company for NICE technology appraisals. International Journal of Technology Assessment in Health Care. 2020;36: 311–316. doi:10.1017/S0266462320000422
26. Augustovski F, Iglesias C, Manca A, Drummond M, Rubinstein A, Marti SG. Barriers to generalizability of health economic evaluations in latin america and the caribbean region. Pharmacoeconomics. 2009;27: 919–29. doi:10.2165/11313670-000000000-00000
27. Arnold RJG, Ekins S. Time for cooperation in health economics among the modelling community. PharmacoEconomics. 2010;28: 609–613. doi:10.2165/11537580-000000000-00000
28. Garcia-Mochon L, Rovira Forns J, Espin J. Cost transferability problems in economic evaluation as a framework for an european health care and social costs database. Cost Effectiveness and Resource Allocation. Springer; 2021;19: 1–12.
29. Pouwels X, Sampson CJ, Arnold RJG. Opportunities and barriers to the development and use of open source health economic models: A survey. Value Health. 2022;25: 473–479. doi:10.1016/j.jval.2021.10.001
30. Sampson CJ, Wrightson T. Model registration: A call to action. PharmacoEconomics - Open. 2017;1: 73–77. doi:10.1007/s41669-017-0019-2
31. Eddy DM, Hollingworth W, Caro JJ, Tsevat J, McDonald KM, Wong JB. Model transparency and validation: A report of the ISPOR-SMDM modeling good research practices task force-7. Med Decis Making. 2012;32: 733–43. doi:10.1177/0272989x12454579
32. European Organization For Nuclear Research, OpenAIRE. Zenodo [Internet]. CERN; 2013. doi:10.25495/7GXK-RD71
33. Quantitative Social Science I for. Dataverse [Internet]. Harvard University; 2007. Available: https://dataverse.org
34. Büyükkaramikli NC, Rutten-van Mölken MPMH, Severens JL, Al M. TECH-VER: A verification checklist to reduce errors in models and improve their credibility. PharmacoEconomics. 2019;37: 1391–1408. doi:10.1007/s40273-019-00844-y
35. Eric Wong W, Debroy V, Choi B. A family of code coverage-based heuristics for effective fault localization. Journal of Systems and Software. 2010;83: 188–208. doi:https://doi.org/10.1016/j.jss.2009.09.037
36. Martin RC. Agile software development: Principles, patterns, and practices. Prentice Hall PTR; 2003.

37. github. GitHub [Internet]. 2007. Available: https://github.com/

38. R Core Team. R: A language and environment for statistical computing [Internet]. Vienna, Austria: R Foundation for Statistical Computing; 2022. Available: https://www.R-project.org/

39. Pan M, Gawthrop PJ, Cursons J, Crampin EJ. Modular assembly of dynamic models in systems biology. PLoS computational biology. Public Library of Science San Francisco, CA USA; 2021;17: e1009513.

40. Wilson JAC Greg AND Bryan. Good enough practices in scientific computing. PLOS Computational Biology. Public Library of Science; 2017;13: 1–20. doi:10.1371/journal.pcbi.1005510

41. Foundation TFS. What is copyleft? [Internet]. Available: https://www.gnu.org/copyleft/

42. Jenkins DA, Martin GP, Sperrin M, Riley RD, Debray TPA, Collins GS, et al. Continual updating and monitoring of clinical prediction models: Time for dynamic prediction systems? Diagnostic and Prognostic Research. 2021;5: 1. doi:10.1186/s41512-020-00090-3

43. Ye Y, Barapatre S, Davis MK, Elliston KO, Davatzikos C, Fedorov A, et al. Open-source software sustainability models: Initial white paper from the informatics technology for cancer research sustainability and industry partnership working group. J Med Internet Res. 2021;23: e20028. doi:10.2196/20028

44. Beck K, Beedle M, Van Bennekum A, Cockburn A, Cunningham W, Fowler M, et al. Manifesto for agile software development. Snowbird, UT; 2001;

45. Kokol P, Blažun Vošner H, Kokol M, Završnik J. Role of agile in digital public health transformation. Frontiers in Public Health. 2022;10. doi:10.3389/fpubh.2022.899874

46. Mergel I. Open collaboration in the public sector: The case of social coding on GitHub. Government Information Quarterly. 2015;32: 464–472. doi:https://doi.org/10.1016/j.giq.2015.09.004

47. Shahin M, Ali Babar M, Zhu L. Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices. IEEE Access. 2017;5: 3909–3943. doi:10.1109/ACCESS.2017.2685629

48. Zhou J, Walker RJ. API deprecation: A retrospective analysis and detection method for code examples on the web. Proceedings of the 2016 24th ACM SIGSOFT international symposium on foundations of software engineering. 2016. pp. 266–277.

49. Chambers JM. Object-Oriented Programming, Functional Programming and R. Statistical Science. Institute of Mathematical Statistics; 2014;29: 167–180. doi:10.1214/13-STS452

50. Barros C, Luo Y, Chubaty AM, Eddy IM, Micheletti T, Boisvenue C, et al. Empowering ecological modellers with a PERFICT workflow: Seamlessly linking data, parameterisation, prediction, validation and visualisation. Methods in Ecology and Evolution. Wiley Online Library; 2023;

51. Statistical Computing RF for. The comprehensive r archive network [Internet]. 2022. Available: https://cran.r-project.org

52. Codecov [Internet]. Available: https://about.codecov.io/

53. Hamilton MP, Gao CX, Filia KM, Menssink JM, Sharmin S, Telford N, et al. Mapping psychological distress, depression and anxiety measures to AQoL-6D utility using data from a sample of young people presenting to primary mental health services. medRxiv. Cold Spring Harbor Laboratory Press; 2022; doi:10.1101/2021.07.07.21260129

54. Hamilton M, Gao C. youthvars: Describe and Validate Youth Mental Health Datasets [Internet]. Zenodo; 2022. doi:10.5281/zenodo.6084467

55. Hamilton M, Gao C. Scorz: Score questionnaire item responses [Internet]. Zenodo; 2022. doi:10.5281/zenodo.6084824

56. Hamilton M, Gao C. specific: Specify Candidate Models for Representing Mental Health Systems [Internet]. Zenodo; 2022. doi:10.5281/zenodo.6116701

57. Gao C, Hamilton M. TTU: Implement Transfer to Utility Mapping Algorithms [Internet]. Zenodo; 2022. doi:10.5281/zenodo.6130155
58. Hamilton MP, Gao CX. Youthu: Transform youth outcomes to health utility predictions [Internet]. Zenodo; 2022. doi:10.5281/zenodo.6210978
59. Hamilton MP. Synthetic (fake) youth mental health datasets and data dictionaries [Internet]. Harvard Dataverse; 2021. doi:10.7910/DVN/HJXYKQ
60. Hamilton MP, Gao CX, Filia KM, Menssink JM, Sharmin S, Telford N, et al. Transfer to AQoL-6D Utility Mapping Algorithms [Internet]. Harvard Dataverse; 2021. doi:10.7910/DVN/DKDIB0
61. Hamilton M, Gao C. Complete study program to reproduce all steps from data ingest through to results dissemination for a study to map mental health measures to AQoL-6D health utility [Internet]. Zenodo; 2022. doi:10.5281/zenodo.6212704
62. Hamilton M, Gao C. aqol6dmap_use: Apply AQoL-6D Utility Mapping Models To New Data [Internet]. Zenodo; 2022. doi:10.5281/zenodo.6416330
63. Hamilton MP. aqol6dmap_fakes: Generate fake input data for an AQoL-6D mapping study [Internet]. Zenodo; 2022. doi:10.5281/zenodo.6321821
64. Hamilton M. ttu_mdl_ctlg: Generate a template utility mapping (transfer to utility) model catalogue [Internet]. Zenodo; 2022. doi:10.5281/zenodo.6116385
65. Hamilton MP. ready4-dev/ttu_lng_ss: Create a Draft Scientific Manuscript For A Utility Mapping Study [Internet]. Zenodo; 2022. doi:10.5281/zenodo.5976988
66. Hugo: The world's fastest framework for building websites [Internet]. Available: https://gohugo.io
67. Docsy [Internet]. Available: https://www.docsy.dev
68. Docsearch [Internet]. Available: https://docsearch.algolia.com
69. Netlify [Internet]. Available: https://www.netlify.com
70. Hamilton M. ready4pack: Author r packages that extend the Ready4 framework [Internet]. 2022. doi:10.5281/zenodo.5644322
71. Quantitative Social Science I for. Sample data usage agreement [Internet]. 2022. Available: https://support.dataverse.harvard.edu/sample-data-usage-agreement
72. Orygen. ready4: A suite of authoring, modelling and prediction tools for exploring topics in young people's mental health [Internet]. 2022. Available: https://github.com/ready4-dev/
73. Orygen. ready4: Open and modular mental health systems models [Internet]. 2022. Available: https://zenodo.org/communities/ready4
74. Orygen. ready4: Open and modular mental health systems models [Internet]. 2022. Available: https://dataverse.harvard.edu/dataverse/ready4)
75. git. Git [Internet]. Available: https://git-scm.com/
76. Henry L, Wickham H. Lifecycle: Manage the life cycle of your package functions [Internet]. 2021. Available: https://CRAN.R-project.org/package=lifecycle
77. Emerson J, Bacon R, Kent A, Neumann PJ, Cohen JT. Publication of decision model source code: Attitudes of health economics authors. PharmacoEconomics. 2019;37: 1409–1410. doi:10.1007/s40273-019-00796-3
78. Anzt H, Bach F, Druskat S, Löffler F, Loewe A, Renard BY, et al. An environment for sustainable research software in germany and beyond: Current state, open challenges, and call for action. F1000Research. Faculty of 1000 Ltd; 2020;9.
79. Chalmers I, Bracken MB, Djulbegovic B, Garattini S, Grant J, Gülmezoglu AM, et al. How to increase value and reduce waste when research priorities are set. The Lancet. Elsevier; 2014;383: 156–165.

80. Craig P, Di Ruggiero E, Frolich KL, Mykhalovskiy E, White M, Campbell R, et al. Taking account of context in population health intervention research: Guidance for producers, users and funders of research. National Institute for Health Research; 2018;
81. Sampson CJ, Arnold R, Bryan S, Clarke P, Ekins S, Hatswell A, et al. Transparency in decision modelling: What, why, who and how? PharmacoEconomics. 2019;37: 1355–1369. doi:10.1007/s40273-019-00819-z
82. Seibel P. Joe armstrong. Coders at work: Reflections on the craft of programming. Springer; 2009. pp. 205–239.