



Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:

Hamilton, MP;Gao, C;Wiesner, G;Filia, KM;Menssink, JM;Plencnerova, P;Baker, DG;McGorry, PD;Parker, A;Karnon, J;Cotton, SM;Mihalopoulos, C

Title:

A Prototype Software Framework for Transferable Computational Health Economic Models and Its Early Application in Youth Mental Health

Date:

2024-08-01

Citation:

Hamilton, M. P., Gao, C., Wiesner, G., Filia, K. M., Menssink, J. M., Plencnerova, P., Baker, D. G., McGorry, P. D., Parker, A., Karnon, J., Cotton, S. M. & Mihalopoulos, C. (2024). A Prototype Software Framework for Transferable Computational Health Economic Models and Its Early Application in Youth Mental Health. *Pharmacoeconomics*, 42 (8), pp.833-842. <https://doi.org/10.1007/s40273-024-01378-8>.

Persistent Link:

<https://hdl.handle.net/11343/352566>

License:

[CC BY-NC](#)



A Prototype Software Framework for Transferable Computational Health Economic Models and Its Early Application in Youth Mental Health

Matthew P. Hamilton^{1,2} · Caroline Gao^{1,2,3} · Glen Wiesner⁴ · Kate M. Folia^{2,3} · Jana M. Menssink^{2,3} · Petra Plencnerova⁵ · David G. Baker^{2,3} · Patrick D. McGorry^{2,3} · Alexandra Parker^{3,6} · Jonathan Karnon⁷ · Sue M. Cotton^{2,3} · Cathrine Mihalopoulos¹

Accepted: 26 March 2024 / Published online: 20 May 2024
© The Author(s) 2024

Abstract

Summary: We are developing an economic model to explore multiple topics in Australian youth mental health policy. To help make that model more readily transferable to other jurisdictions, we developed a software framework for authoring modular computational health economic models (CHEMs) (the software files that implement health economic models). We specified framework user requirements for: a simple programming syntax; a template CHEM module; tools for authoring new CHEM modules; search tools for finding existing CHEM modules; tools for supplying CHEM modules with data; reproducible analysis and reporting tools; and tools to help maintain a CHEM project website. We implemented the framework as six development version code libraries in the programming language R that integrate with online services for software development and research data archiving. We used the framework to author five development version R libraries of CHEM modules focussed on utility mapping in youth mental health. These modules provide tools for variable validation, dataset description, multi-attribute instrument scoring, construction of mapping models, reporting of mapping studies and making out of sample predictions. We assessed these CHEM module libraries as mostly meeting transparency, reusability and updatability criteria that we have previously developed, but requiring more detailed documentation and unit testing of individual modules. Our software framework has potential value as a prototype for future tools to support the development of transferable CHEMs.

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

✉ Matthew P. Hamilton
matthew.hamilton1@monash.edu

- ¹ School of Public Health and Preventive Medicine, Monash University, Melbourne, Australia
- ² Orygen, Parkville, Australia
- ³ Centre for Youth Mental Health, The University of Melbourne, Parkville, Australia
- ⁴ Heart Foundation, Docklands, Australia
- ⁵ Headspace National Youth Mental Health Foundation, Melbourne, Australia
- ⁶ Institute for Health and Sport, Victoria University, Footscray, Australia
- ⁷ Flinders Health and Medical Research Institute, Flinders University, Adelaide, Australia

Key Points for Decision Makers

How health economic models are implemented as software significantly shapes how easy they are to transfer to other decision contexts.

We have developed a software framework that can help implement a transparent, reusable and updatable economic model in youth mental health.

Our framework can be a prototype for software tools that improve the transferability of health economic models more generally.

1 Introduction

Computational models, particularly those addressing economic topics, have become essential tools for health policy development [1, 2]. Although influential and widely used, health economic models can be skills and resource intensive to develop. One suggestion for making health economic modelling projects more tractable is for health economists to re-use each other's models [3].

Key concerns for health economists when considering whether to reuse a model are generalisability (application without adaptation) and transferability (application of selected components and/or application with modification) [4]. Health economic models are likely to require at least some modification before they can be validly applied to a new jurisdiction, time period or decision problem. Making such modifications is harder if the software files that implement the model – the computational health economic model (CHEM) – cannot be readily understood, used and edited.

Some of the barriers to the development of transferable CHEMs could be addressed by appropriate software frameworks. A software framework is a shared common technology used by developers to collaboratively author software [5]. A software framework provides a foundation for developing multiple software applications with shared resources (e.g. code and data files), that can be modified to suit specific needs. 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 [6].

In this paper, we describe:

- (i) our motivation for developing a transferable CHEM in youth mental health;
- (ii) a prototype software framework we have developed to support the development of transferable CHEMs; and
- (iii) how we have applied the prototype software framework to undertake initial development of our youth mental health CHEM.

2 Motivation

We became interested in solving some of the technical barriers to transferable CHEMs in 2018 when considering how to synthesise multiple youth mental health economic modelling projects in one multi-purpose model. This overarching youth mental health model is called *readyforwhatsnext* [7].

2.1 Economic Topics

We are constructing *readyforwhatsnext* from projects in 4 of the 12 domains of health economics identified by Wagstaff and Culyer [8]. These are:

- health and its value (our projects: models to map psychological and functional measures to health utility);
- determinants of health and ill health (our projects: models for creating synthetic household populations with risk and protective factors for mental disorders);
- demand for health and healthcare (our projects: spatial epidemiology and help-seeking choice models); and
- supply of health services (our projects: a model of primary mental healthcare services for young people).

Potential future directions for *readyforwhatsnext* may incorporate projects in two additional domains: (i) public health (to model the impact of selected fiscal policy and regulation options on young people's mental health); and (ii) human resources (to model the supply and behaviours of the youth mental health workforce). Our ultimate aim is to flexibly combine model components in analyses that answer questions in a further two domains:

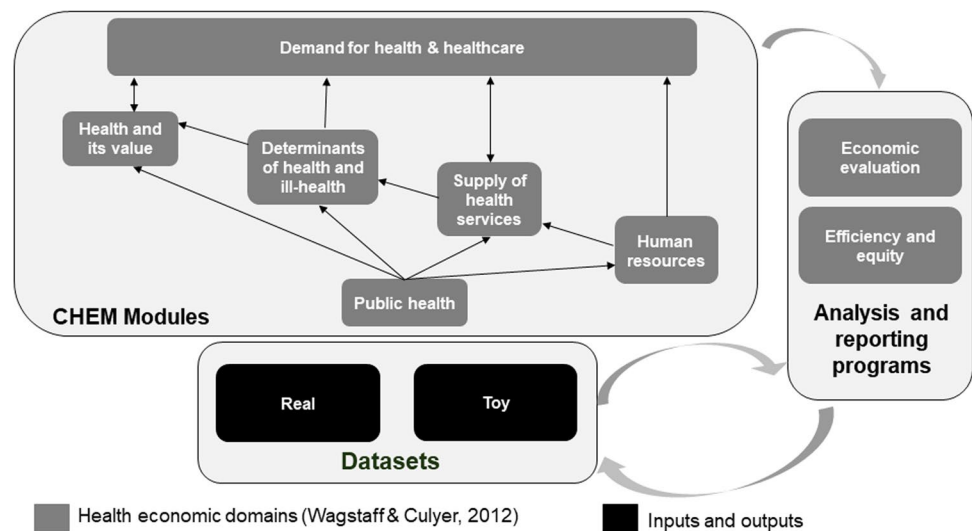
- efficiency and equity (our goal: assess the distributional impacts and identify the optimal targeting of youth mental health interventions); and
- economic evaluation (our goal: assess the value for money and affordability of competing policy options for improving the mental health of young people).

2.2 Transferability

We are principally interested in using *readyforwhatsnext* to answer policy questions relating to the mental health of young people in Australia. Australian public mental health services can be planned and commissioned by the Australian Government, the governments of Australia's state and territories and independent, regionally focussed organisations called Primary Health Networks. The mental health of young people is a global priority [9] and Australia has initiated youth mental health service model reforms that have been adopted in other jurisdictions [10]. For these reasons, we aim to develop *readyforwhatsnext* in a manner that facilitates its transfer to multiple decision contexts.

However, there are ethical risks associated with each potential transfer, in particular the need to ensure the social acceptability, fitness for purpose and appropriate use of a CHEM in each new decision context [11]. We therefore aim

Fig. 1 High-level summary of planned implementation of youth mental health economic model



for readyforwhatsnext to meet six transparency, reusability and updatability (TRU) criteria. We have described these criteria and their origins in ethical modelling practice elsewhere [11], but briefly they are:

- (T1) Software files are open access;
- (T2) Developer contributions and judgments on appropriate use are easily identified;
- (R1) Programming practices promote selective reuse;
- (R2) Licenses permit derivative works;
- (U1) Maintenance infrastructure is in place; and
- (U6) Releases are systematically retested and deprecated.

To meet these criteria we are implementing readyforwhatsnext as an open source, modular model. Open source projects grant third parties permissions to access, use and modify model source code and data. A modular model is constructed from multiple reusable and replaceable sub-models (modules) [12]. When undertaking analyses intended to inform decision making, readyforwhatsnext uses real Australian data (which can be empirical, simulated or assumption-based). To help demonstrate the potential transferability of readyforwhatsnext to other decision contexts, we are creating synthetic or toy datasets (“fake” data that realistically represent the key features of observed data, but which can be shared without confidentiality concerns [13]).

A high-level schematic guiding our implementation of readyforwhatsnext is outlined in Fig. 1. To implement this plan, we first developed a software framework to assist with general (i.e., model agnostic) tasks relating to authoring, management and use of CHEM modules, datasets and analysis and reporting programs. We then used that framework to develop artefacts (e.g., variable validation tools) specific to the distinctive features of the readyforwhatsnext model.

3 Prototype Software Framework

To help readyforwhatsnext meet all six TRU criteria, we developed a prototype software framework called ready4 [14].

3.1 Framework User Requirements

The framework user requirements (FURs) for ready4 evolved over 6 years (2018–2024) as we encountered new implementation issues for readyforwhatsnext that we needed to solve. Currently, these requirements are:

FUR 1: A reusable template for CHEM modules that can be: (i) run as independent models; (ii) safely combined, sharing inputs and outputs with each other without unintended interference; and (iii) selectively replaced, updated or extended.

FUR 2: A programming syntax that promotes simplicity and consistency in how algorithms associated with CHEM modules are: (i) labelled and (ii) used in health economic analysis programs.

FUR 3: Module authoring tools that integrate with online software development services to help author CHEM modules that are: (i) created using the template module; (ii) written in a consistent house style; (iii) always at least minimally documented; (iv) licensed for reuse; (v) bundled as R libraries with authorship, funding source, version number and digital object identifier (DOI) metadata; and (vi) quality assured with publicly available testing artefacts.

FUR 4: Data management tools that integrate with online data repository services to streamline: (i) the ingesting and labelling of confidential and non-confidential data and (ii) the sharing of non-confidential data.

FUR 5: Replication and reporting tools that: help (i) author analysis and reporting template programs for rendering HTML, Adobe PDF and Microsoft Word outputs; (ii) retrieve version-specific analysis and reporting template programs from online code repositories; and (iii) supply analysis and reporting template programs with study specific data (e.g. authorship details, input data and analysis results) to render custom reports.

FUR 6: Search tools to retrieve web-based information on a CHEM project's: (i) model modules and tutorials; (ii) datasets and data collections; and (iii) replication and reporting template programs.

FUR 7: Documentation tools that integrate with open source website development and hosting infrastructure to help: (i) develop a CHEM project documentation website that consolidates content from the websites of individual CHEM module libraries, datasets and programs and (ii) partially automate the updating of website resources such as tutorials and itemised lists of code and data releases.

3.2 Framework Implementation

We implemented ready4 as R [15] code libraries that integrate with a number of online services. The framework is documented by a project website.

3.2.1 R Libraries

R libraries typically depend on other R libraries. As the number of dependencies of an R library grows, so does the fragility of that library (e.g. the library may cease to work as intended due to changes in one of its dependency libraries). To reduce the fragility of our framework we implemented it as multiple R libraries rather than one R library. We authored six novel R libraries to implement the ready4 framework, all of which have distinct purposes and dependencies (Table 1).

To meet FUR 1, the ready4 [16] foundation library defines a template CHEM module using R's S4 class system. Benefits of using the S4 class system include automated verification that data supplied to modules adhere to pre-specified criteria and support for object-oriented programming. Two notable features of object-oriented programming are encapsulation [17] and inheritance [17]. Encapsulation allows a CHEM module to bundle model data and algorithms in a protected environment, which is useful for helping ensure that CHEM modules continue to work as intended when combined. Inheritance allows new CHEM modules to inherit some or all of the properties of a parent module. Inheritance is useful when modifying CHEM modules, for example, to account for: (i) interaction effects when CHEMs are combined or (ii) distinctive features of new decision contexts to which they are being transferred. The ready4 template

Table 1 Software framework R libraries

Library	Purpose	R library dependencies
ready4	Provide a template and novel syntax for modular CHEM implementations and tools for finding interoperable CHEM modules, datasets and reproducible analysis programs	dataverse dplyr gh kableExtra lifecycle magrittr methods piggyback purrr rlang rvest stats stringr tibble tidyRSS tidyselect tools utils
ready4fun	Streamline and standardise the authoring and documenting of functions that support transferable and generalisable model algorithms.	desc devtools dplyr gert Hmisc lifecycle lubridate magrittr methods piggyback pkgdown purrr readxl ready4 ready4show ready4use rlang sinew stats stringr stringr testit tibble tidytools usethis utils xfun
ready4class	Streamline and standardise the authoring and documenting of new interoperable CHEM modules	devtools dplyr fs gtools Hmisc knitr lifecycle magrittr methods purrr ready4 ready4fun ready4show rlang stats stringr stringr testit testthat tibble tidy usethis utils
ready4pack	Help bundle and disseminate newly created CHEM modules as R libraries that are documented, licensed and quality assured	dataverse dplyr knitr lifecycle magrittr methods purrr ready4 ready4class ready4fun rlang stringr testthat tibble tidyr utils
ready4use	Help manage the labelling and transfer of data between CHEM modules and local and remote data repositories	data.table dataverse dplyr fs Hmisc knitr lifecycle magrittr methods piggyback purrr readxl ready4 ready4show rlang stats stringr stringr testit testthat tibble tidyr utils
ready4show	Facilitate the use of CHEM modules in programs to implement integrated and reproducible data ingest, analysis and reporting pipelines	dataverse DescTools dplyr flextable grDevices here Hmisc kableExtra knitr knitrBootstrap lifecycle magrittr methods officer purrr ready4 rlang rmarkdown stringr stringr testthat tibble tidyr utils xtable

CHEM module can be used to create other CHEM modules that inherit a common set of properties. One of these inherited properties is a novel syntax of 15 core commands [18] that enable CHEM module algorithms to be consistently named and described (meeting FUR 2). The ready4 library also contains tools for retrieving web-based information on CHEM modules, datasets and analysis programs (meeting FUR 6) and tools for developing and maintaining a CHEM project's documentation website (meeting FUR 7).

Three module authoring libraries are designed to meet FUR 3. The ready4fun [19] library contains tools for writing functions in a consistent house style and automatically generating basic documentation for each function. The ready4class [20] library contains tools to help streamline and standardise the authoring of novel, documented CHEM modules from the template CHEM module. The ready4pack [21] library provides tools for disseminating themed bundles of CHEM modules as R libraries that are: documented (with a website and PDF manuals); licensed (using the copyleft GNU GPL-3 [22] by default); easily citable (citation information can be retrieved within an R session or from hosting repositories); and quality assured (each update triggers the running of any tests created by module library authors, with test results publicly documented).

The ready4use [23] library for managing CHEM data contains tools for supplying CHEM modules with input data ingested from local (i.e. a user's computer) or remote (online repositories) locations, labelling CHEM module datasets and exporting CHEM module data to online repositories (meeting FUR 4). The ready4show [24] library for authoring reproducible analysis programs and reporting subroutines (FUR5) contains tools to help write executables that apply CHEM modules to compatible datasets for the purpose of undertaking reproducible health economic analyses. The library supports the creation of analysis programs that are self-documenting (code is integrated with plain English explanations of what it does) and subroutines that can be used as templates that trigger the creation of explanatory documents (e.g. a scientific manuscript).

3.2.2 Online Services

Framework libraries are designed to be used in conjunction with a number of online services. The module authoring libraries use the GitHub [25] service's tools for: software development, testing, version control and distribution; hosting of documentation websites; and transparent record keeping of collaborator contributions. Continuous integration [26] tools provided by GitHub are used to trigger systematic retesting of module libraries with every edit. Tests can be unit tests (verifying the correct output when small, isolated sections of code are run independently) and

acceptance tests (verifying the correct output from multiple code components are run together to meet core user requirements [27]). The module authoring libraries use the codecov [28] service to measure and share information about code coverage [29]—the proportion of code that has been explicitly tested. The open science repository service Zenodo [30] is used by the module authoring libraries to provide persistent storage, uniquely identified with a digital object identifier (DOI), of each code release. The ready4use library uses the Dataverse [31] data repository service for persistent, uniquely identifiable and versioned storage of model input and output data. For the management of frequently updated datasets that do not require persistent storage, all framework libraries use GitHub.

3.2.3 Documentation Website

A project documentation website (<https://www.ready4-dev.com>) provides guidance to model developers on how to use and contribute improvements to the ready4 software framework. Prior versions of the documentation website are archived and publicly accessible. The documentation website was developed using the Hugo framework [32], Docsy theme [33] and Algolia DocSearch [34] and is hosted using the Netlify [35] service. We linked our Netlify account to our GitHub organisation so that the project website automatically updates whenever its (publicly available) source code is edited.

4 Prototype Model Modules

Our early use of the ready4 framework has focussed on developing readyforwhatsnext modules for constructing, sharing and using utility mapping models in youth mental health.

4.1 Model User Requirements

The model user requirements (MURs) that we specified for the initial modules of readyforwhatsnext are:

MUR 1: Variable validation tools that: (i) define the allowable values and numeric class (integer or double precision) associated with different psychological, functional and health utility measures appropriate for use in youth mental health samples and (ii) provide an informative error message that halts an analysis or reporting program's execution should impermissible values be assigned to each measure.

MUR 2: Dataset description tools that: (i) attach structured metadata to a human records dataset that specify unique person identifier, group assignment and data collection round variables and (ii) generate tabular and graphical summaries of dataset descriptive statistics.

MUR 3: Multi-attribute instrument scoring tools that: (i) attach structured metadata about an instrument (e.g. name, version, country, items, domains, parameter values for scoring algorithms) to a human records dataset; (ii) apply pre-written (for AQoL-6D [36] and EQ-5D [37] instruments) or user-supplied instrument scoring functions to a human records dataset; and (iii) generate descriptive plots of a dataset's instrument scores for individual items, domains and totals (weighted and unweighted).

MUR 4: Utility mapping model construction tools that: (i) attach structured metadata about the target utility instrument and candidate predictor variables and covariates to be used in constructing utility mapping models to a human records dataset; (ii) generate descriptive tables, a correlation matrix and plots about the variables to be assessed for inclusion in models; (iii) attach structured metadata about the candidate utility mapping models to be assessed; (iv) generate tabular and graphical summaries of the performance of each combination of model, predictors and covariates specified for evaluation by either a user or by a default algorithm; and (v) strip confidential records from the constructed models so that they are able to be publicly shared.

MUR 5: Utility mapping study reporting tools that: (i) generate from a reporting template a PDF catalogue summarising the predictive performance under multiple configurations of utility mapping models selected for public dissemination; (ii) generate from a reporting template (either a pre-written default or a user-supplied customisation) a scientific manuscript summarising a utility mapping study; (iii) share model catalogues and (if desired) scientific summary along with metadata about utility mapping models in an open access repository; and (iv) facilitate the authoring of a consolidated study replication program that implements data ingest, instrument scoring, model construction, report authoring and dissemination of study artefacts.

MUR 6: Utility prediction tools that: (i) search open access repositories for utility mapping models developed with readyforwhatsnext; (ii) retrieve relevant metadata, including a link to the model catalogue for models specified by a user; (iii) apply selected utility mapping models to make out of sample predictions using a dataset provided by a user; and (iv) (where there is health utility data at two timepoints) convert predicted health utility scores to quality adjusted life years (QALYs).

4.2 Model Implementation

4.2.1 Online Service Configuration

We established and configured accounts with the online services supported by ready4. We created a GitHub

organisation (a collection of code repositories) where source code that we author is stored and version controlled at <https://github.com/ready4-dev/>. We configured individual repositories in our GitHub organisation to implement continuous integration to assess the compliance of our module libraries with policies specified by the Comprehensive R Archive Network (CRAN) [38]. To facilitate the creation and hosting of module library websites, we enabled GitHub Pages in each code repository. We also created a Zenodo community; a collection of permanent, uniquely identified repositories, at <https://zenodo.org/communities/ready4/>. 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 archived on Zenodo with a DOI. Finally, to manage model datasets, we created a dedicated collection (<https://dataverse.harvard.edu/dataverse/ready4>) within the Harvard Dataverse installation.

4.2.2 Module Libraries

We used all six ready4 framework libraries to author five development version readyforwhatsnext module libraries (Table 2). These libraries are `youthvars` [39] which addresses MURs 1 and 2, `scorz` [40] which addresses MUR 3, `specific` [41] which addresses MUR 4, `TTU` [42] which addresses MUR 5 and `youthu` [43] which addresses MUR 6.

4.2.3 Reporting Templates

We used the `ready4show` library to author two reporting templates that are designed to be used in conjunction with these utility mapping modules. The first reporting template [44] is used to create a catalogue of utility mapping models. The second reporting template [45] can be used to generate a manuscript providing a scientific summary of each study implemented with these modules.

4.2.4 Application

We have previously described a study [46] that used these module libraries and reporting templates to develop and document utility mapping models in a sample of young people presenting to primary mental health services. We used these module libraries to write programs for replicating that study's reporting and analysis algorithm [47], to demonstrate how to apply the utility mapping models constructed in that study to new data [48] and to generate a synthetic representation of the study dataset [49]. We also created an open access study input parameters and results dataset [50] and toy datasets to help demonstrate the transferability of the study algorithm [51].

Table 2 Model module libraries for utility mapping

Library	Purpose	R library dependencies
youthvars	Describe and validate youth mental health human record datasets	arsenal assertthat car cowplot dplyr ggplot2 ggpubr gridExtra here Hmisc hutils kableExtra knitr knitrBootstrap lifecycle lubridate magrittr MASS Matrix matrixcalc methods mice psych purrr ready4 ready4show ready4use rlang scales simstudy stats stringi stringr testthat tibble tidyselect utils
scorz	Score multi-attribute instruments	dplyr eq5d Hmisc hutils knitr knitrBootstrap lifecycle magrittr methods mice purrr ready4 ready4use rlang simstudy stats stringi stringr testthat tibble tidyselect youthvars
specific	Construct and evaluate utility mapping models	assertthat boot Boruta brms caret cowplot dataverse DescTools dplyr enrichwith ggalt ggfortify ggplot2 gtools here Hmisc hutils kableExtra knitr knitrBootstrap lifecycle magrittr MASS methods parallel psych purrr randomForest ready4 ready4show ready4use rlang scorz stats stringi stringr synthpop testthat tibble tidyr tidyselect utils viridis xfun youthvars
TTU	Implement reproducible utility mapping studies	dplyr ggplot2 Hmisc knitr lifecycle magrittr methods purrr R.utils ready4 ready4show ready4use rlang scorz specific stringr testthat utils
youthu	Apply utility mapping models to out of sample data	assertthat BCEA boot dataverse dplyr lifecycle lubridate magrittr MatchIt methods purrr ready4 ready4use rlang specific stats stringi stringr tibble tidyr tidyselect truncnorm utils youthvars

4.3 Documentation Website

Using the framework website as a template, we used the search and documentation tools from the ready4 library to create and maintain a project documentation website (<https://readyforwhatsnext.org/>). The project website includes details on available model modules, datasets and analysis programs and tutorials on how these artefacts can be used.

4.4 Transferability Assessment

We subjectively assessed these utility mapping model artefacts against TRU criteria (Table 3). For each criterion, we provided a global assessment of whether it was met using the responses ‘yes’, ‘no’ or ‘partial’. We believe that our utility mapping study module libraries, code and data repositories and documentation have satisfactorily met four of the six criteria (T1, R1, R2 and U1) and have partially met two criteria (T2 and U2). The main shortcomings we identified were that individual modules and functions have yet to: (i) be supported with human authored documentation that includes examples of how they can be used and (ii) be quality assured through unit testing.

5 Discussion

We have developed a software framework and used it to author CHEM modules that largely satisfy explicit criteria for transparency, reusability and updatability. We have used toy data to demonstrate how these CHEM module libraries, originally developed for a single utility mapping study [46], can be applied: (i) to datasets from new samples

with different variable names; (ii) to construct models with different utility instruments, predictors and covariates; and (iii) to develop and use both cross-sectional and longitudinal mapping models.

Our framework and CHEM module R libraries are currently publicly available as development releases. We plan to undertake a number of key improvements, such as more detailed documentation, comprehensive unit testing and streamlining dependencies, prior to submitting these libraries for quality assurance and dissemination by CRAN [38].

Although designed to address challenges specific to our project, ready4 can be a prototype for future software frameworks that meet the user requirements of other modelling teams and projects. Our prototype has a number of features that such future work can build on. Firstly, developing a software framework to work within an existing and widely used open source programming language such as R can keep framework scope relatively narrow. This makes it more tractable to develop, maintain and learn, and facilitates integration with other modelling tools written in that language (e.g. the dependency libraries we list in Tables 1 and 2). Secondly, a sensible trade-off needs to be found between transparent code implementation (which requires clear and sufficiently detailed documentation) and prioritising the development of working code over writing documentation (a foundation principle of Agile Software Development [52]). Our software framework makes this trade off by enforcing the use of consistent code style conventions and file organisation which in turn enables automated generation of simple documentation at every CHEM module update.

Our framework addresses barriers to transferability specific to how a model is implemented as software (i.e. CHEMs). Health economists must also consider conceptual

Table 3 Assessment of a CHEM implementation in youth mental health using transparent, reusable and updatable (TRU) criteria

Criterion	Met?	Detail
T1 Software files are open access	Yes	All source code and testing procedures are available in public GitHub repositories, with each code release persistently available in a Zenodo repository. As the study dataset contains confidential patient records, it was not published. Instead, a synthetic representation of the study dataset is persistently available in a repository in the Harvard Dataverse. Data files to support out of sample application of models are published at the same location
T2 Developer contributions and judgements on appropriate use are easily identified	Partial	All module libraries, programs and datasets are distributed with citation information. Public GitHub repositories detail author code contributions over project development history. Model catalogues persistently available on the Harvard Dataverse describe the predictive performance of models under multiple usage regimes. Each code library is documented with worked examples of how to apply CHEM modules. However, documentation of individual functions and methods are algorithm generated and need more customisation, including more examples of use. Analysis and reporting programs are self-documenting. Sub-routines for generating reports are documented with README files
R1 Programming practices promote selective reuse	Yes	CHEMs are written using both functional and object-oriented paradigms. Code library documentation websites include hypothetical examples of generalisability (applying study algorithm to estimate mapping models from new data with the same predictor and outcome variables) and transferability (adapting study algorithm to develop mapping models from datasets with different predictor variables and outcomes measured with a different utility instrument)
R2 Licenses permit derivative works	Yes	All code is distributed using GPL-3 licenses. Datasets use an amended version of template terms provided by Harvard Dataverse, allowing reuse of data subject to some ethical restrictions (e.g. use in efforts to reidentify study participants is prohibited)
U1 Maintenance infrastructure is in place	Yes	All code is version controlled using git and GitHub, with semantic versioning. Each code library has a specified maintainer, and guidance for potential code contributors is available on the project website
U2 Releases are systematically re-tested and deprecated	Partial	Continuous integration is used for all code libraries, primarily for acceptance testing. Only limited use is made of unit testing. Retired library code is deprecated using tools from the lifecycle R library. Library documentation articles and datasets are also deprecated

or theoretical validity. For example, the candidate models currently available in our scorz library [40] were chosen because of their suitability for mapping to the utility instrument (AQoL-6D [36]) used in the study [46] for which scorz was originally developed. We have identified a need to use alternative models for mapping to utility instruments that produce less continuous total score distributions [46]. Our software framework is designed to make such extensions more straightforward to implement.

Factors such as usability, active user-community and supporting resources influence adoption of software frameworks [5]. To systematically address these issues for our prototype would require investments in: (i) engaging the health economic modelling community in refining framework user requirements; (ii) undertaking software re-development and testing to meet those requirements; (iii) incorporating the evolving code and documentation authoring capabilities of large language models; (iv) the development of improved and more comprehensive documentation and training resources;

and (v) proactive user-community building and support. We currently lack the resources to undertake these tasks.

Adequately planned, resourced and implemented software frameworks are potentially important enablers of improved health economic modelling practice. Reference models [53], methodological innovation to improve model transferability [54] and the development of regularly updated CHEMs that can support ‘living Health Technology Assessment’ [55] would all be facilitated by a trusted software framework for implementing more transparent, reusable and updatable CHEMs. A future software framework for CHEMs would ideally incorporate a base set of features useful to developers of computational models across all domains of public health and health economics, with the capability for community-led extensions that are tailored to the needs of modellers focussed on specific health-conditions. Such approaches will likely be important to ensure a sufficient potential user-base to make CHEM software frameworks sustainable.

6 Conclusion

Our software framework provides a solution to some technical barriers we faced when seeking to implement a transferable CHEM in youth mental health. Our framework can be used as a prototype for developing future software frameworks that address the user requirements of the broader health economic modelling community.

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

Declarations

Funding Open Access funding enabled and organized by CAUL and its Member Institutions. Software framework development was funded by Orygen, VicHealth, Victoria University and an Australian Government Research Training Program (RTP) Scholarship to Matthew Hamilton. The utility mapping study to which utility mapping modules were applied was funded by the National Health and Medical Research Council (NHMRC, APP1076940), Orygen and headspace.

Conflict of interest None declared.

Ethics approval Software framework development did not involve human subject research and was not ethically reviewed. The utility mapping modules were originally developed to implement 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).

Availability of data and materials The ready4 framework is documented at <https://www.ready4-dev.com>. Documentation on readyforwhatsnext is available at <https://readyforwhatsnext.org/>. 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>.

Author contributions M.P.H. led the conceptualisation and software implementation of ready4 and readyforwhatsnext and manuscript drafting. M.P.H., C.G. and G.W. co-authored the software described in this article. All co-authors critically reviewed and revised manuscript drafts and approved the final manuscript text.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License, which permits any non-commercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc/4.0/>.

References

1. 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 Econ*. 2015;24(10):1256–71.
2. Erdemir A, Mulugeta L, Ku JP, Drach A, Horner M, Morrison TM, et al. Credible practice of modeling and simulation in health-care: ten rules from a multidisciplinary perspective. *J Transl Med*. 2020;18(1):369. <https://doi.org/10.1186/s12967-020-02540-4>.
3. Arnold RJG, Ekins S. Time for cooperation in health economics among the modelling community. *Pharmacoeconomics*. 2010;28(8):609–13. <https://doi.org/10.2165/11537580-000000000-00000>.
4. 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(11):919–29. <https://doi.org/10.2165/11313670-000000000-00000>.
5. Myllärniemi V, Kujala S, Raatikainen M, Sevoñ P. Development as a journey: factors supporting the adoption and use of software frameworks. *J Softw Eng Res Dev*. 2018;6(1):1–22.
6. Edwin NM. Software frameworks, architectural and design patterns. *J Softw Eng Appl*. 2014;2014.
7. readyforwhatsnext development team. readyforwhatsnext—a prototype modular and open-source economic model of youth mental health. <https://readyforwhatsnext.org/>.
8. Wagstaff A, Culyer AJ. Four decades of health economics through a bibliometric lens. *J Health Econ*. 2012;31(2):406–39.
9. McGorry PD, Mei C, Chanan A, Hodges C, Alvarez-Jimenez M, Killackey E. Designing and scaling up integrated youth mental health care. *World Psychiatry*. 2022;21(1):61–76. <https://doi.org/10.1002/wps.20938>.
10. Hetrick SE, Bailey AP, Smith KE, Malla A, Mathias S, Singh SP, et al. Integrated (one-stop shop) youth health care: best available evidence and future directions. *Med J Aust*. 2017;207(10):S5–18. <https://doi.org/10.5694/mja17.00694>.
11. Hamilton MP, Gao C, Karnon J, Salvador-Carulla L, Cotton SM, Mihalopoulos C. Ethical considerations when planning, implementing and releasing health economic model software: a new proposal. 2024. arXiv preprint [arXiv:2403.17798](https://arxiv.org/abs/2403.17798).
12. Pan M, Gawthrop PJ, Cursons J, Crampin EJ. Modular assembly of dynamic models in systems biology. *PLoS Comput Biol*. 2021;17(10): e1009513.
13. Bellovin SM, Dutta PK, Reitering N. Privacy and synthetic datasets. *Stanf Technol Law Rev*. 2019;22:1.
14. ready4 development team. ready4—A prototype software framework for living, transferable health economic models. <https://ready4-dev.com/>.
15. R Core Team. R: A Language and Environment for Statistical Computing. Vienna. <https://www.R-project.org/>.
16. Hamilton M. ready4: implement modular and open-source health economic models. R package version 0.1.6.9000. <https://doi.org/10.5281/zenodo.5606250>. <https://ready4-dev.github.io/ready4/>.
17. Hourani H, Wasmi H, Alrawashdeh T. A code complexity model of object oriented programming (OOP). In: 2019 IEEE Jordan international joint conference on electrical engineering and information technology (JEEIT). 2019. p. 560–64.
18. ready4 development team. Syntax—ready4 modules use a simple and consistent syntax. <https://www.ready4-dev.com/docs/getting-started/implementation/syntax/>.
19. Hamilton M, Wiesner G. ready4fun: author and document functions that implement transferable health economic model algorithms. R package version 0.0.0.9535. <https://doi.org/10.5281/zenodo.5611778>. <https://ready4-dev.github.io/ready4fun/>.

20. Hamilton M, Wiesner G. ready4class: author and document classes that make health economic models modular. R package version 0.0.0.9296. <https://doi.org/10.5281/zenodo.5640313>. <https://ready4-dev.github.io/ready4class/>.
21. Hamilton M. ready4pack: author R packages of health economic model modules. R package version 0.0.0.9033. <https://doi.org/10.5281/zenodo.5644322>. <https://ready4-dev.github.io/ready4pack/>.
22. Free Software Foundation. Licenses. <https://www.gnu.org/licenses>.
23. Hamilton M, Wiesner G. ready4use: author, ingest, label and share health economic model datasets. R package version 0.0.0.9231. <https://doi.org/10.5281/zenodo.5644336>. <https://ready4-dev.github.io/ready4use/>.
24. Hamilton M, Wiesner G. ready4show: author health economic analysis programs and reporting templates. R package version 0.0.0.9111. <https://doi.org/10.5281/zenodo.5644568>. <https://ready4-dev.github.io/ready4show/>.
25. github. GitHub. <https://github.com/>.
26. Shahin M, Ali Babar M, Zhu L. Continuous integration, delivery and deployment: a systematic review on approaches, tools, challenges and practices. *IEEE Access*. 2017;5:3909–43. <https://doi.org/10.1109/ACCESS.2017.2685629>.
27. Martin RC. Agile software development: principles, patterns, and practices. Prentice Hall PTR; 2003.
28. Codecov. <https://about.codecov.io/>. Accessed 26 Jan 2023.
29. Eric Wong W, Debroy V, Choi B. A family of code coverage-based heuristics for effective fault localization. *J Syst Softw*. 2010;83(2):188–208. <https://doi.org/10.1016/j.jss.2009.09.037>.
30. European Organization For Nuclear Research, OpenAIRE. Zenodo. CERN. <https://www.zenodo.org/>.
31. Institute For Quantitative Social Science. Dataverse. Harvard University. <https://dataverse.org>.
32. Hugo. The world's fastest framework for building websites. <https://gohugo.io>. Accessed 26 Jan 2023.
33. Docsy. <https://www.docsy.dev>. Accessed 26 Jan 2023.
34. Docsearch. <https://docsearch.algolia.com>. Accessed 19 Apr 2023.
35. Netlify. <https://www.netlify.com>. Accessed 26 Jan 2023.
36. Richardson JR, Peacock SJ, Hawthorne G, Iezzi A, Elsworth G, Day NA. Construction of the descriptive system for the assessment of quality of life AQoL-6D utility instrument. *Health Qual Life Outcomes*. 2012;10(1):38.
37. Devlin NJ, Brooks R. EQ-5D and the EuroQol group: past, present and future. *Appl Health Econ Health Policy*. 2017;15:127–37.
38. R Foundation. The Comprehensive R Archive Network. <https://cran.r-project.org>.
39. Hamilton M, Gao C. youthvars: describe and validate youth mental health datasets. R package version 0.0.0.9125. <https://doi.org/10.5281/zenodo.5646550>. <https://ready4-dev.github.io/youthvars/>.
40. Hamilton M, Gao C. scorz: score multi-attribute utility instruments. R package version 0.0.0.9065. <https://doi.org/10.5281/zenodo.5722708>. <https://ready4-dev.github.io/scorz/>.
41. Hamilton M, Gao C. specific: specify models to solve inverse problems. R package version 0.0.0.9108. <https://doi.org/10.5281/zenodo.5768689>. <https://ready4-dev.github.io/specific/>.
42. Gao C, Hamilton M. TTU: implement transfer to utility mapping algorithms. R package version 0.0.0.9366. Zenodo. <https://doi.org/10.5281/zenodo.5646593>. <https://ready4-dev.github.io/TTU/>.
43. Hamilton MP, Gao CX. youthu: transform youth outcomes to health utility predictions. R package version 0.0.0.9122. <https://doi.org/10.5281/zenodo.5646668>. <https://ready4-dev.github.io/youthu/>.
44. Hamilton M. ttu_md1_ctlg: generate a template utility mapping (transfer to utility) model catalogue. Version 0.1.0.0. <https://doi.org/10.5281/zenodo.5936870>.
45. Hamilton MP. ready4-dev/ttu_Ing_ss: create a draft scientific manuscript for a utility mapping study. Version 0.9.0.1. <https://doi.org/10.5281/zenodo.5976987>.
46. 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*. 2022. <https://doi.org/10.1101/2021.07.07.21260129>.
47. 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. Version 0.1. <https://doi.org/10.5281/zenodo.6116077>.
48. Hamilton M, Gao C. aqol6dmap_use: apply AQoL-6D utility mapping models to new data. Version 0.1. <https://doi.org/10.5281/zenodo.6317179>.
49. Hamilton MP. aqol6dmap_fakes: generate fake input data for an AQoL-6D mapping study. Version 0.0.9.0. <https://doi.org/10.5281/zenodo.6321821>.
50. Hamilton MP, Gao CX, Filia KM, Menssink JM, Sharmin S, Telford N, et al. Transfer to AQoL-6D utility mapping algorithms. Harvard Dataverse. <https://doi.org/10.7910/DVN/DKDIB0>.
51. Hamilton MP. Synthetic (fake) youth mental health datasets and data dictionaries. Harvard Dataverse. <https://doi.org/10.7910/DVN/HJXYKQ>.
52. Beck K, Beedle M, van Bennekum A, Cockburn A, Cunningham W, Fowler M, et al. Manifesto for agile software development. <http://www.agilemanifesto.org/>.
53. Afzali HH, Karnon J, Merlin T. Improving the accuracy and comparability of model-based economic evaluations of health technologies for reimbursement decisions: a methodological framework for the development of reference models. *Med Decis Mak*. 2013;33(3):325–32. <https://doi.org/10.1177/0272989x12458160>.
54. Craig P, Di Ruggiero E, Frohlich K, 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. *Online of NIHR Journals Library*. National Institutes of Health Research; 2018.
55. Thokala P, Srivastava T, Smith R, Ren S, Whittington MD, Elvidge J, et al. Living health technology assessment: issues, challenges and opportunities. *Pharmacoeconomics*. 2023;41(3):227–37.