Developing flexible, iterative, and transparent decision models

A detailed look at a rheumatoid arthritis individual patient simulation

April 18, 2018
Agenda

• Characteristics of relevant decision models
• Motivating example: the IVI-RA simulation model
• Developing models using R
• Creating web interfaces
• Lessons learned and tools for future model development
Characteristics of relevant decision models
Mathematical models play a central role in value assessment

• Frequent need for extrapolations beyond the time-horizon, interventions, outcomes, and settings observed in the available evidence

• Mathematical models are used to combine the findings of different studies to calculate the benefits, risks, and cost of interest for value assessment

• Idiosyncratic choices regarding model structure have an impact on findings

• Lack of transparency

• Cumbersome, if not impossible, for someone other than the original model developer to update the analysis
Barriers to the use of model-based value assessment by decision-makers

- Published model-based CEAs perceived as complex; difficult to judge credibility
- Published CEAs are quickly outdated given pace of new clinical evidence
- Published CEAs may have limited relevance given the local context
- Limited tools to facilitate setting-specific value assessments
- Time constraints
Reducing barriers to decision-making by developing models that are:

- **Flexible**: Tailored to the specific population and disease area of interest, and with sufficient capability for sensitivity/uncertainty analysis
- **Iterative**: Updated over time as new evidence emerges
- **Transparent**: Easily “pressure tested” by experts and understood by decision-makers
Motivating example: the IVI-RA simulation model
Overview of the IVI-RA model

- Open source decision model to assess the value of different (sequences of) conventional and targeted disease-modifying anti-rheumatic drugs (DMARDs) for the treatment of rheumatoid arthritis (RA)
  - CEA and MCDA
- Discrete-time individual patient simulation with 6 month cycles
- Accounts for both parameter and structural uncertainty
- Model input parameters based on the literature
- Competing model structures were informed by existing cost-effectiveness models and clinical expertise
Backbone: Development of HAQ over time
Competing model structures

300+ model structures

• Short-term effect of treatment on HAQ

• Causes of treatment switching

• Long-term progression of HAQ

• The probability distribution used to measure time to treatment discontinuation

• Algorithm used to simulate utility
Alternative structural assumptions regarding the impact of treatment on HAQ and switching
Long term progression of HAQ over time

- Constant rate of progression
- Latent class growth model
  - Different subgroups have distinct HAQ trajectories and the rate of worsening of HAQ progression decreases over time
Duration of maintenance treatment

7 possible parametric distributions: exponential, Weibull, Gompertz, gamma, lognormal, log-logistic, and generalized gamma
Utility

![Graph showing the relationship between Mean utility and HAQ, with two lines representing different models: Alava (2013) mixture model and Wailoo (2006).]
## Model source data - literature based

<table>
<thead>
<tr>
<th></th>
<th>RCT evidence</th>
<th>Routine practice, observational evidence</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treatment effects at 6 months</td>
<td>ACR/ DAS28/ HAQ</td>
<td>Mapping between endpoints</td>
<td></td>
</tr>
<tr>
<td>Long term treatment effects (6+ months)</td>
<td></td>
<td>HAQ trajectory over time</td>
<td></td>
</tr>
<tr>
<td>Treatment duration</td>
<td></td>
<td>Corrona</td>
<td></td>
</tr>
<tr>
<td>Adverse events</td>
<td>Serious infections</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Utility</td>
<td></td>
<td>HAQ -&gt; EQ-5D</td>
<td></td>
</tr>
<tr>
<td>Mortality</td>
<td></td>
<td>• US life tables</td>
<td>Drug regimen according</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Impact of HAQ on mortality</td>
<td>FDA label</td>
</tr>
<tr>
<td>Resource use</td>
<td></td>
<td>• Physician visits,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Chest X-rays tuberculosis tests,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• outpatient visit</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• HAQ -&gt; hospitalization</td>
<td></td>
</tr>
<tr>
<td>Productivity</td>
<td></td>
<td>HAQ -&gt; productivity</td>
<td></td>
</tr>
</tbody>
</table>
Model outcomes

- Clinical outcomes during initial treatment phase: ACR response, EULAR response, DAS28, SDAI, CDAI
- Long-term clinical outcomes: HAQ, QALYs
- Adverse events: number of serious infections
- Health care sector costs: drug acquisition and administration costs, general management and monitoring costs, adverse event costs, hospitalization costs
- Non-health care sector costs: productivity losses
- Value assessment: CEA and MCDA
Simulation

• Simulation structure
  ▪ Discrete time individual patient simulation (IPS)
  ▪ Probabilistic sensitivity analysis (PSA)
  ▪ Structural uncertainty analysis

• Computationally intensive
  ▪ Run 10,000 individual patients
  ▪ Assume all patients survive 30 years (60 model cycles)
  ▪ Sample 1,000 parameter sets for PSA
  ▪ Consider 50 model structures
  ▪ => 30 billion iterations
Developing models using R
IVI-RA model development process

1. Parameter estimation using R’s packages for statistical analysis
2. Development of simulation using R and C++ (Rcpp)
3. Analysis of model output using R
4. Web applications with R Shiny
Why R?

- Vast array of statistical packages for parameter estimation (e.g., *flexsurv*, *rjags*) and simulation (e.g., random sampling functions)
- Easy to speed up computationally intensive code by linking R to C++ with Rcpp
- Facility for dynamic report generation (e.g., *knitr*)
- Script based languages are reproducible and transparent
- Automated code testing
Using R packages to develop models

Bundles together relevant components:

- Data
- Code
- Documentation
- Testing

Easy to share with others either as an open-source package or privately
IVI-RA package directory structure

- **data-raw**: Raw data and all statistical analysis scripts to produce model inputs (reproducible via a Makefile)
- **data**: Model inputs created using scripts in data-raw
- **docs**: Model documentation including package website and PDF technical document
- **R**: Code for functions needed to run the model with R
- **src**: C++ code for the IPS. Linked to R with Rcpp
- **tests**: Unit testing via R package testthat
Speeding up models with Rcpp

- Individual patient simulations are notoriously slow, especially when combined with probabilistic sensitivity analysis
  - Using an IPS based on the diabetes UKPDS-68 outcomes equations, McEwan et al. (2010) showed that a PSA with VBA took 13.5 hours while the same simulation with C++ took 3 minutes
- Rcpp simplifies object interchange between R and C++
- C++ code easily added within R packages

```cpp
#include <Rcpp.h>

//@export
//@[Rcpp::export]
double C_sum(std::vector<double> x){
    double total = 0;
    for(int i = 0; i < x.size(); ++i) {
        total += x[i];
    }
    return total;
}

> x <- c(1, 2, 3)
> C_sum(x)
[1] 6
```

Version control

• Version control helps model developers track all modifications to code over time
• Allows developers to revert to old versions or to create new “branches” in order to add new features or experiment
• The code history for the IVI-RA model is available on GitHub
• Models released can be versioned on GitHub (or on CRAN as an R package)
Unit testing

• The IVI-RA package contains hundreds of unit tests to help ensure that the code works as intended

• Each time the code is pushed (e.g., updated) to GitHub:
  • It is re-compiled and installed on an external Ubuntu machine with Travis-CI
  • codecov.io estimates the percent of the code that is covered by the tests

```r
\# exponential distribution
def fit <- flexsurv::flexsurvreg(Surv(futime, fustat) ~ age, data = ovarian, dist = "exp")

fit.lrate <- fit$coef[1]*x
set.seed(50)
samp1 <- rexp(n, rate = exp(fit.lrate))
set.seed(50)
samp2 <- replicate(n, iviRA::rsurvC(fit.lrate, anci = 0, dist = "exponential"))
expect_equal(samp1, samp2)
```
Documentation with R packages

Website with pkgdown

Dynamic report with knitr

8.5 HAQ trajectory with tDMARD maintenance treatment

Based on the NDB longitudinal study, Wolfe and Michaud (2010) estimated the overall annual HAQ progression rate among RA patients who had switched to biologic treatment at -0.001 (95CI: -0.004 to 0.002). In a separate analysis, also based on NDB data, Michaud et al. (2011) reported annual HAQ progression rates by treatment adjusted for baseline HAQ score, age, sex, education, smoking, BMI, comorbidity, and RA onset. The average HAQ rate among patients on a biologic was -0.001 as well, which instills confidence that the reported HAQ progression rates for different biologies as reported by Michaud et al. (2011) can be directly compared with the overall annual HAQ progression rate of 0.031 reported by Wolfe and Michaud (2010). Accordingly, biologic specific HAQ progression rates by Michaud et al. (2011) are used in the model. For tDMARD treatments
Creating web interfaces
Why do we need web interfaces?

- Translates complex models into a simpler, graphical form
- Health economists and modelers can use or evaluate models without specialized knowledge or access to a particular programming language or software platform
- Can help facilitate feedback from non-technical stakeholders and use by decision-makers
- Allows model developers to use tools fit for purpose
Allowing advanced users to modify the IVI-RA model

- Users can modify the characteristics of the population, select up to 5 treatment sequences of arbitrary length, select a model structure (below), and modify nearly all model parameters.
In the cost-effectiveness table, the ICER is a single value; however, in truth, there is considerable uncertainty around this number. The cost-effectiveness plane is a common way to assess this uncertainty. The cloud of points are from separate simulations and reflect the range of possible outcomes based on scientific uncertainty. A treatment is cost-effective with high probability if the majority of points lie to the right of the red willingness to pay line (which is based on the willingness to pay value chosen above).

Conditional on users choice of WTP per QALY

The value of the selected treatment sequences can also be assessed by assigning a monetary value to a QALY. Total benefits reflect the monetized value of QALYs less the costs associated with treatment. This plot displays this monetized value, which is referred to as the net monetary benefit. The value assigned to the QALY is based on the willingness to pay value chosen above. The treatment sequence with the greatest net monetary benefit can be considered the most cost-effective.
Using R Shiny to develop web interfaces

- shiny is an R package that allows modelers to build interactive web interfaces straight from R
- Allows modelers to develop web interfaces relatively quickly using only the R programming language
- Can be highly customized using JavaScript, HTML, and CSS
- OpenCPU is an alternative solution for R-based models that allows even more customization than Shiny
Lessons learned and tools for future model development
Lessons learned

• Define your objectives
  • Does your model need to be continuously updated over time?
  • How fast does your model need to be?
  • Who is the intended audience?
  • How general does your code need to be?

• Plan before coding
  • Plan your code base carefully before you begin coding (e.g., wait until you have a defined model structure based on the available evidence base)
  • Don’t begin building a web app until you know what your target audience is looking for

• Don’t optimize code too early
  • If an optimization (e.g., speed improvement) isn’t trivially clear, don’t optimize until you can profile your code
Tools for future model development

• Currently developing an open-source R package for simulating and analyzing models with R

• Some features that may be useful
  • Generalized framework for using statistical models and patient-level data to create model structures at R level
  • Support for common models such as partitioned survival analysis and state-transition modeling (cohort, IPS), written in C++ under the hood for speed
  • Summarizing probabilistic sensitivity analyses (also look at the bcea R package)
  • Relevant functions for survival distributions at C++ level (exponential, Weibull, Gompertz, gamma, lognormal, log-logistic, generalized gamma, splines, fractional polynomials) for cases in which there is a need to create bespoke IPS models and performance is important
Q&A session

Thank you for your attention!