# Prostate-Predict – initial requirements

I’ll start by concentrating on the tool, but I’ve included some thoughts on the wider site.

## Tool

### Model

For the model, the stata code is acceptable. If there are documentation and equations that would help me understand it, so much the better. In any case, I’d like to be able to run the model so I can compare its output against the output of a web version of the model. We can spot check by hand, but if a command line version is possible - taking inputs, treatments and generating results - then we can automate things and get better test coverage.

A working stata code for individual mock cases is available. This automatically creates a graph which can be removed (lines 93-122). The relevant outputs are ‘pred\_PC\_cum’ and ‘pred\_NPC\_cum’ (predicted Prostate cancer (and Non-Prostate Cancer) cumulative mortality). [Please ignore the overall survival, PCSM and NPCM columns which are a ‘work-around’ to create the stacked area chart]

I’m assuming that an R model isn’t available? If it were we might consider serving it out directly through an API.

No sadly it is in R only, and I’m not really competent in R…

### A list of inputs. For each input, we’ll need:

* A widget type.  In the breast cancer version, we only use numeric inputs or radio button selections, but others are possible if they help. We replaced check boxes with radio button yes/no selections to keep a consistent interface.
* The associated variable used in the stata code and the values it can take.
* For numeric inputs, minimum and maximum values, precision, and step size.
* For radio buttons, the names used in the stata code for the allowed values, and their likely external names on the buttons.
* A label for the widget.
* Units and conversions as necessary. (e.g. metric/british for BMI)
* Whether the value may be unknown. In breast cancer predict, we require an entry for all widgets before we display results, but for most inputs this can be ‘Unknown’.
* Some help text. This appears in a popup window when the user presses the info button associated with the widget.

Inputs for Prostate:

Numeric:

-Age (18-100) (integers) (years)

-PSA (0-100) (any value to 1d.p) (ng/ml)

Fixed:

-T stage (1, 2, 3, 4) (compulsory information)

-Grade Group (1, 2, 3, 4, 5) (compulsory information) ((Coud have a ‘converter’ for Gleason to grade group)

-Biopsy (0, 1, 2) (0=unknown(default), 1=<50%cores involved, 2=≥50% cores involved) ((could have a X/Y biopsy cores converter)

-‘Comorbidity’ (0 or 1) ((1=hospital admission in preceding 2 years with Charlson score 1+- needs info))

Treatment:

-(0 or 1). (0=’Conservative management’ 1=’Radical treatment’) Default to 0.

There are no interactions between inputs.

Are there any interactions between inputs? e.g. in breast cancer predict, the metastases widget is greyed out unless the number of positive nodes is zero.

We’ll need to lay the widgets out on the page somehow, so if you have order or grouping preferences that would be useful to know. The powerpoint slides may cover this.

### Treatments

In PREDICT breast cancer, treatments are cumulative and ordered (hormone, then chemo, then trastuzumab), and some may be enabled or disabled according to the values of the inputs. (e.g. hormone therapy is disabled if the ER status is negative). Again, we need associated help text, variables and values, defaults and labelling.

Treatment is relatively simple in the Prostate model. This should default to 0 (Conservative management/no treatment) with ‘1’ (radical treatment) shown separately. 2 coours can be used on the surviva curves as per the breast prototype. Treatment ‘3’ (ADT) should be ignored.

### Results

For results, I suspect you may want similar options of a Table, Curves, Bar charts, Texts and Icons. It’s probably best to go through the result tabs on the breast cancer tool and suggest revisions as necessary for prostate. See <https://wintoncentre.maths.cam.ac.uk/predict> username: predict, password: feedback.

Yes I think replicating these options would be ideal. My preference would be for the Bar chart or Curve to be the default presentation, but we should really go through a structured patient/clinician feedback process on this Prostate model. For now, it’s reasonable to go with the Breast analysis.

This breast cancer tool is currently still missing a good print option but we are working on this. You will probably have some thoughts on what should be in a printable prostate version.

### Adverse Effects

We are working on a rapid review for the adverse effects of predict breast cancer treatments, and expect to present these on the site in due course. Something similar would be good for prostate treatments.

Links to informational sites can be inserted – we will need to find suitable versions for modern UK men.

# The wider site

* Is the site to be open access like the current predict?
* For the broader web site, we’d need more texts, any relevant images, links to other relevant sites etc. The current breast cancer development site should be a good reference for these, though this is still incomplete.
* Disclaimers, Privacy policy, Cookie warnings, Legal (copyright and licensing). Private or public source code?
* Branding and home pages and linkages between predict branded sites will need further discussions between all involved of course.
* Menu structure

These will need some discussion. Whether it can be open access will depend slightly on Jem and others. Mmost likely it should be password-protected during initial testing. The Academic Urology Group will need to be cited in some capacity (and maybe Prof Pharoah’s Cancer Genetic Epidemiology group…/Funding bodies…)

## Updates and maintenance

I’d recommend keeping implementations of individual predict sites separate as this will make future updates and maintenance easier. We can rely on linking and branding to create the impression of a coherent PREDICT branded umbrella site.

The tool in the existing predict breast cancer site is coded in javascript. The model and coefficients and user interface are all hard coded there. The new interface currently follows a similar route, but we are compiling into javascript from clojurescript source code.

In future we should move the data structures driving the model, the user interfaces and the texts to a database where they can be updated independently of the codebase and so make it unnecessary to rebuild the web site whenever anything changes.

Agree with all the above. As discussed, a mechanism whereby we could modify text independently would be ideal.