Recent Infection Test Calibration (MDRI and FRR)

Eduard Grebe

2016-05-04

This vignette covers the use of functions `mdrical` and `frrcal`.

Introduction

Incidence estimates from cross-sectional surveys using biomarkers for 'recent infection' require that the test for recent infection (usually an adapted diagnostic assay) be accurately characterised. The two critical parameters of test performance are the Mean Duration of Recent Infection (MDRI), denoted Ω_T, (with T the recency cutoff time), and False Recent Rate (FRR), denoted β_T. The explicit time cutoff T was introduced by Kassanjee et al. *Epidemiology*, 2012.[1] to differentiate between 'true recent' and 'false recent' results. Also see Kassanjee, McWalter, Welte. *AIDS Research and Human Retroviruses*, 2014.[2]They state:

> To lead to an informative estimator, this cut-off, though theoretically arbitrary, must be chosen to reflect the temporal dynamic range of the test for recent infection; i.e. at a time T post infection, the overwhelming majority of infected people should no longer be testing "recent", and furthermore, T should not be larger than necessary to achieve this criterion.[3]

MDRI is defined as the average time alive and returning a 'recent' result, while infected for times less than T. FRR is defined as the proportion of subjects returning a 'recent' result while infected for longer than T.

Test performance may be context-specific, and therefore, where available, local data should be used to calibrate tests. However, should published estimates be used, these may need to be adapted to the local context. Often cross-sectional incidence surveys incorporate recency testing using a Recent Infection Testing Algorithms (RITAs) and it is important to realise that the entire RITA must be appropriately calibrated. This may involve adapting MDRI estimates for more sensitive screening tests (depending on the case definition of 'recent'), or adapting FRR estimates based on weighted estimates from specimen subsets appropriate to the local population (e.g. large numbers of treated individuals). Where possible calibration should be performed using the same set of biomarkers used in a RITA, such as by including a viral load threshold in the calibration step.

Estimating MDRI using binomial regression

This package provides the function `mdrical` to estimate MDRI for a given biomarker or set of biomarkers from a dataset of based on the test being applied to well-characterised specimens and subjects. That is, time since 'infection' should be well-known, as well as test result(s). Note that 'infection' can be arbitrarily defined as the reference time (e.g. the exposure event, date of first detectability on an RNA assay, Western Blot seroconversion, etc.) but should be consistently used. If the reference time used in test calibration differs from the screening assay or algorithm that is used define someone as HIV-positive in a RITA, MDRI needs to be appropriately adapted to cater for this difference.

[1]Kassanjee, R., McWalter, T.A., Baernighausen, T. and Welte, A. "A new general biomarker-based incidence estimator." Epidemiology; 2012, 23(5): 721-728; doi:%5B10.1097/EDE.0b013e3182576c07%5D(http://dx.doi.org/10.1097/EDE.0b013e3182576c07).

[2]Kassanjee, R., McWalter, T.A. and Welte, A. "Short Communication: Defining Optimality of a Test for Recent Infection for HIV Incidence Surveillance." AIDS Research and Human Retroviruses; 2014, 30(1): 45-49; doi:%5B10.1089/aid.2013.0113%5D(http://dx.doi.org/10.1089/aid.2013.0113); PubMed.

[3]Kassanjee, R., McWalter, T.A., Baernighausen, T. and Welte, A. "A new general biomarker-based incidence estimator." Epidemiology; 2012, 23(5): 721-728; doi:%5B10.1097/EDE.0b013e3182576c07%5D(http://dx.doi.org/10.1097/EDE.0b013e3182576c07).

mdrical estimates MDRI by fitting a model for the probability of testing 'recent' as a function of time since infection $P_R(t)$. As an option, one of two functional forms (parameterisations) can be selected by the user. Fitting is performed using a generalised linear model (as implemented in the *glm2* package) to estimate parameters, with two separate link functions, the complementary log-log link or the logit link.

The linear binomial regression model takes the following form, with $g()$ the link function

$$g(P_R(t)) = f(t) \tag{1}$$

If the argument *functional_forms* is specified with the value *"cloglog_linear"*, $g()$ is the complementary log-log link function and $\ln(t)$ as linear predictor of $P_R(t)$, so that:

$$\ln\left(-\ln(1 - P_R(t)\right) = \beta_0 + \beta_1 \ln(t) \tag{2}$$

If the argument `functional_forms` is specified with the value `"logit_cubic"`, $g()$ is the complementary log-log link function and the linear predictor of $P_R(t)$ is a cubic polynomial in t, so that:

$$\ln\left(\frac{P_R(t)}{1 - P_R(t)}\right) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 \tag{3}$$

In both cases, MDRI is the integral of $P_R(t)$ from 0 to T.

$$\Omega_T = \int_0^T P_R(t)dt \tag{4}$$

The default behaviour is to implement both model forms if the argument *functional_forms* omitted.

Confidence intervals are computed by means of subject-level bootstrapping. Because measurements from subjects with more than one measurement in the dataset cannot be considered indpendent observations, subjects (rather than data points) are resampled, with replacement. An MDRI estimate is then computed using the resampled data. The number of bootstraps is specified using the argument *n_bootstraps*. We recommend 10,000 for reproducable confidence intervals and standard errors. It is further necessary to identify the subject identifier in the dataset using the `subid_var` argument.

It is necessary to specify the value of T (using the argument `recency_cutoff_time`) and a time exclusion rule (i.e. to exclude data points beyond a certain time so that falsely recent measurements do not affect the fit between 0 and T unduly) using the argument `inclusion_time_threshold`. This should typically be a value somewhat (but not too much) larger than T. Remember to always use the same unit of time in all options.

You can either supply a list of variables and thresholds (indicating in whether a result above or below the thresholds signify recency) or specify the `recency_rule` as `"binary_data"`, in which case you need a variable with a 1 for recent results and a 0 for non-recent results.

Example of `mdrical` using the complementary log-log functional form and pre-classified data

Load the package in order to use it

```
library(inctools)
```

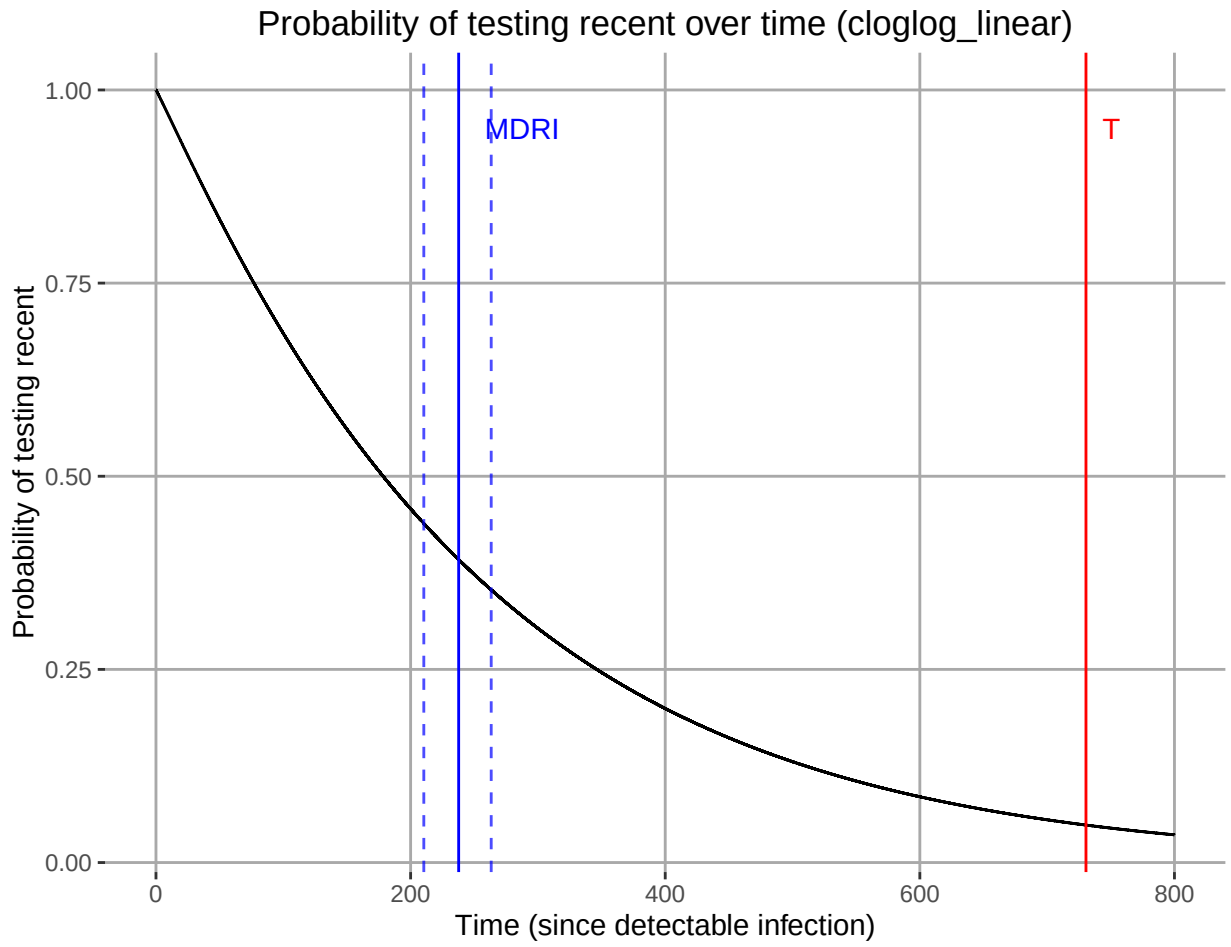
```
exampledata <- read.csv("../data/exampledata_testcalibration.csv")
```

In the first example, we will use a variable that has pre-classified results are recent or non-recent. We will use the complementary log-log functional form only.

Note: To keep compute time reasonable during execution of the example code, only 1,000 bootstraps are performed. We recommend 10,000 to get reasonable standard errors and confidence intervals.

```
mdrical(data=exampledata,
                 subid_var = "SubjectID",
                 time_var = "DaysSinceEDDI",
                 recency_cutoff_time = 730.5,
                 inclusion_time_threshold = 800,
                 functional_forms = c("cloglog_linear"),
                 recency_rule = "binary_data",
                 recency_vars = "Recent",
                 n_bootstraps = 1000,
                 alpha = 0.05,
                 plot = TRUE)
```

```
## $MDRI
##                       PE   CI_LB    CI_UB      SD
## cloglog_linear 237.7463 210.372 263.3076 13.4948
##
## $Plots
## $Plots$cloglog_linear


##
##
## $Models
## $Models$cloglog_linear
##
## Call:  glm2::glm2(formula = (1 - recency_status) ~ 1 + I(log(time_since_eddi)),
##     family = binomial(link = "cloglog"), data = data, control = glm.control(epsilon = tolerance,
##         maxit = maxit, trace = FALSE))
##
## Coefficients:
##             (Intercept)  I(log(time_since_eddi))
##                  -5.786                   1.045
##
## Degrees of Freedom: 707 Total (i.e. Null);  706 Residual
## Null Deviance:       941.2
## Residual Deviance: 714   AIC: 718
```



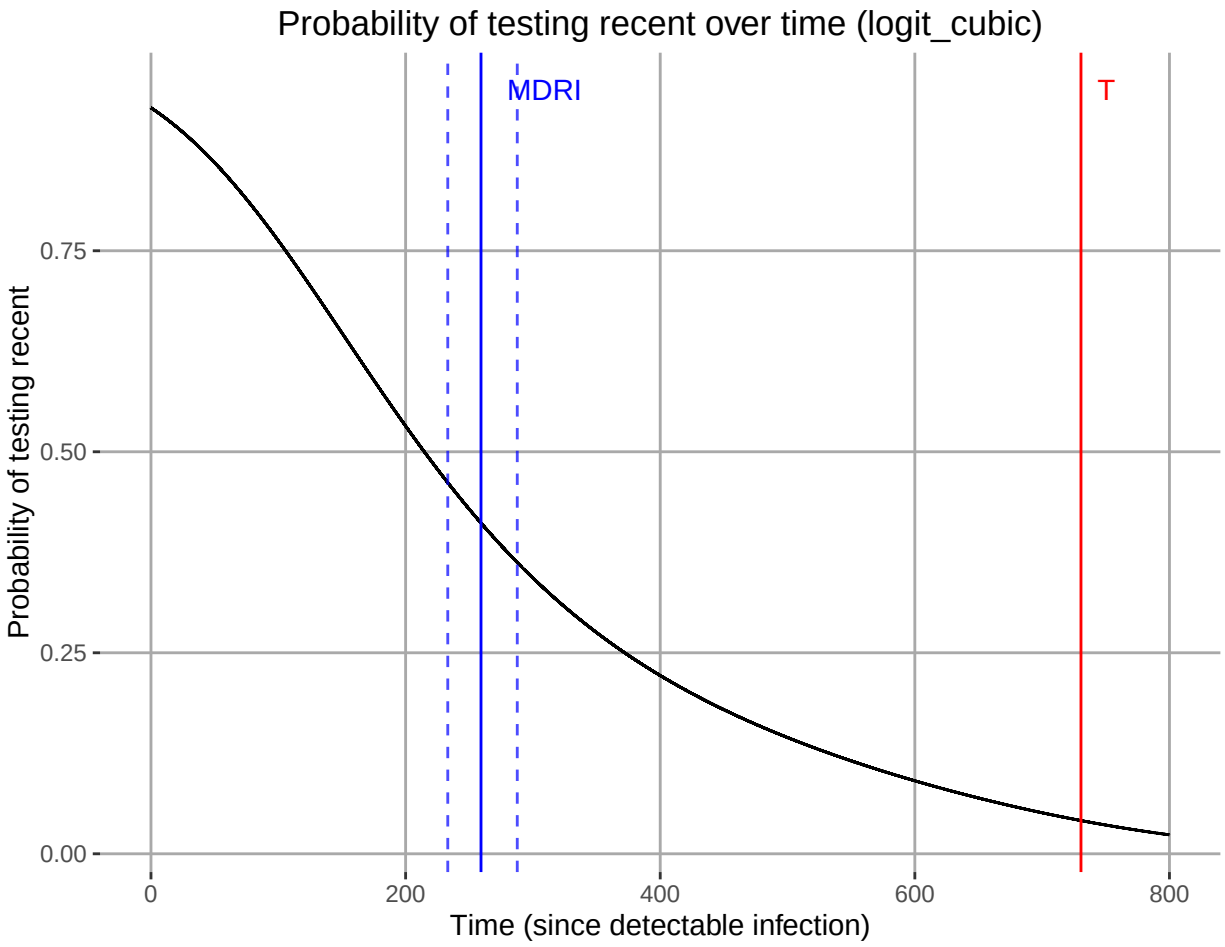
Example of `mdrical` using the both functional forms and two independent thresholds on biomarkers

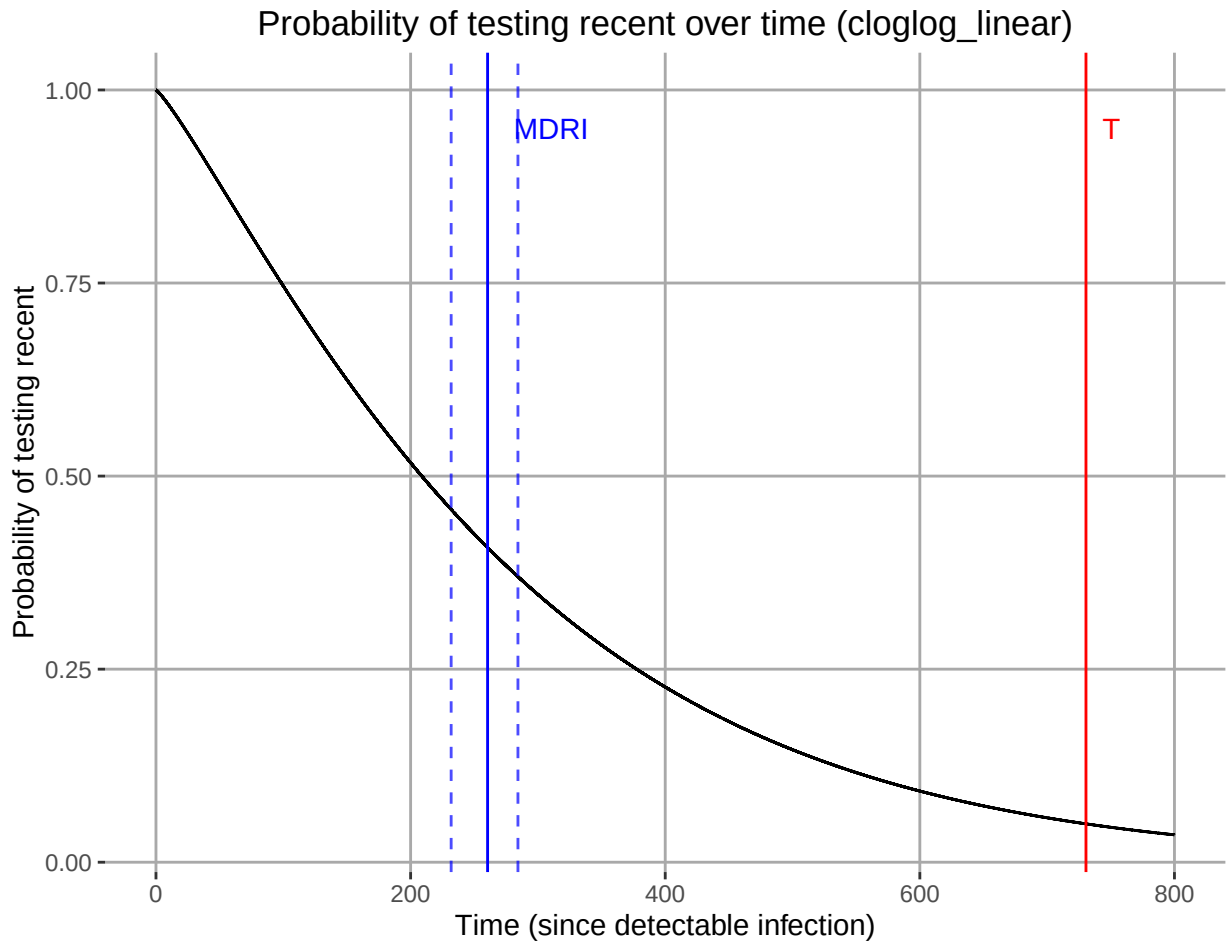
Here we are also specifying a vector of variables and a vector of paramaters to define recency In this case we are using the assay result and the viral load. The paramaters in the vector *c(10,0,1000,1)* mean that recency is defined as an assay biomarker reading below 10 and a viral load reading above 1000.

```
mdrical(data=exampledata,
                subid_var = "SubjectID",
                time_var = "DaysSinceEDDI",
                recency_cutoff_time = 730.5,
                inclusion_time_threshold = 800,
                functional_forms = c("logit_cubic","cloglog_linear"),
                recency_rule = "independent_thresholds",
                recency_vars = c("Result","VL"),
                recency_params = c(10,0,1000,1),
                n_bootstraps = 1000,
                alpha = 0.05,
                plot = TRUE)
```

```
## $MDRI
##                     PE    CI_LB    CI_UB      SD
```

```
## logit_cubic    259.2234 233.1254 287.7184 14.4312
## cloglog_linear 260.4783 231.8452 284.3198 13.7418
##
## $Plots
## $Plots$logit_cubic

##
## $Plots$cloglog_linear

##
##
## $Models
## $Models$logit_cubic
##
## Call:  glm2::glm2(formula = recency_status ~ 1 + I(time_since_eddi) +
##     I(time_since_eddi^2) + I(time_since_eddi^3), family = binomial(link = "logit"),
##     data = data, control = glm.control(epsilon = tolerance, maxit = maxit,
##         trace = FALSE))
##
## Coefficients:
##          (Intercept)    I(time_since_eddi)  I(time_since_eddi^2)
##            2.554e+00           -1.591e-02             2.184e-05
## I(time_since_eddi^3)
##           -1.471e-08
##
## Degrees of Freedom: 643 Total (i.e. Null);  640 Residual
## Null Deviance:       875.9
## Residual Deviance: 635.3     AIC: 643.3
##
## $Models$cloglog_linear
##
## Call:  glm2::glm2(formula = (1 - recency_status) ~ 1 + I(log(time_since_eddi)),
##     family = binomial(link = "cloglog"), data = data, control = glm.control(epsilon = tolerance,
##         maxit = maxit, trace = FALSE))
##
## Coefficients:
##             (Intercept)  I(log(time_since_eddi))
##                  -6.618                   1.170
##
## Degrees of Freedom: 643 Total (i.e. Null);  642 Residual
## Null Deviance:       875.9
## Residual Deviance: 637.5     AIC: 641.5
```


Probability of testing recent over time (logit_cubic)
MDRI
T
0.75
0.50
0.25
0.00
Probability of testing recent
0
200
400
600
800
Time (since detectable infection)



Example of `mdrical` in which bootstraps are run in parallel

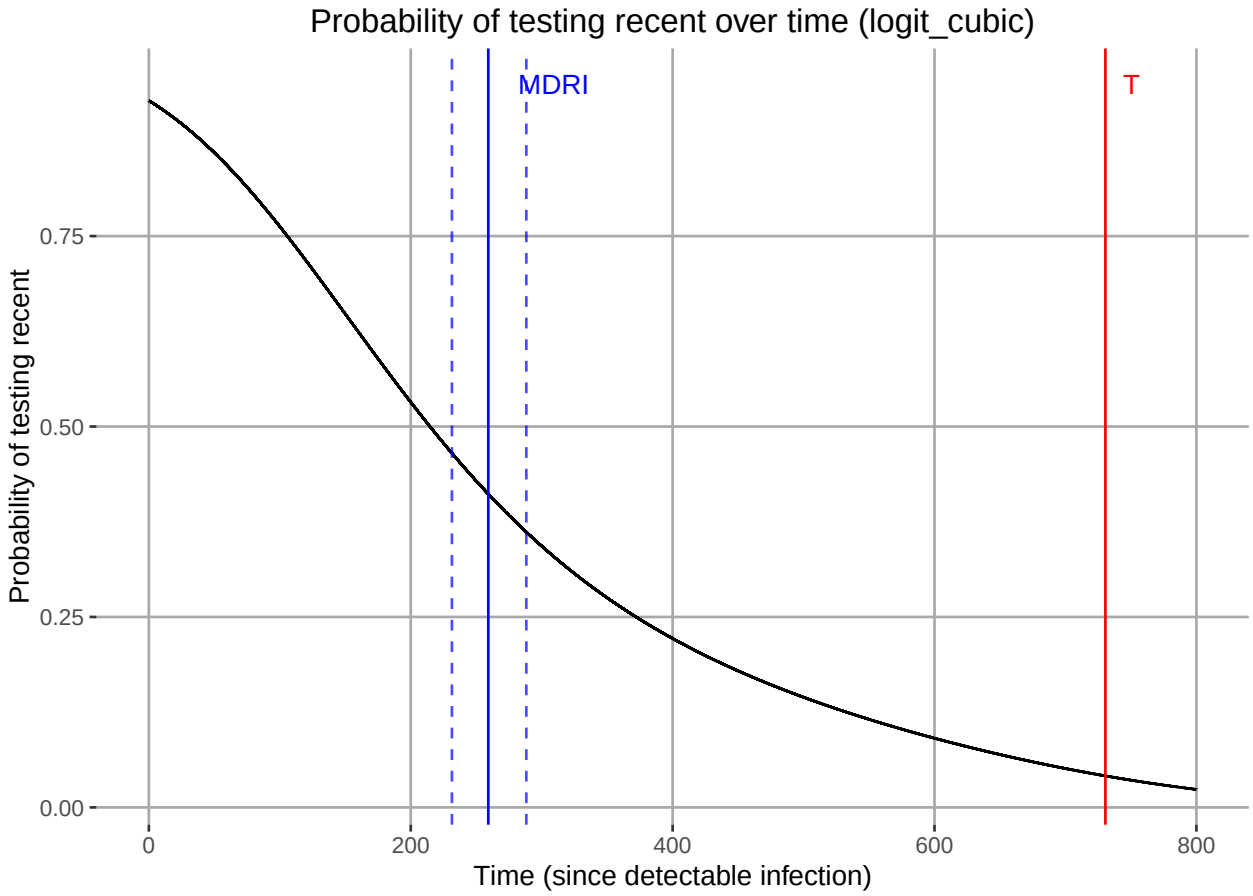
As above, but parellelise the bootstrapping. In this case, split the job over four cores.

Note: This only works on Unix (Mac or Linux). On Windows this will result in error messages.

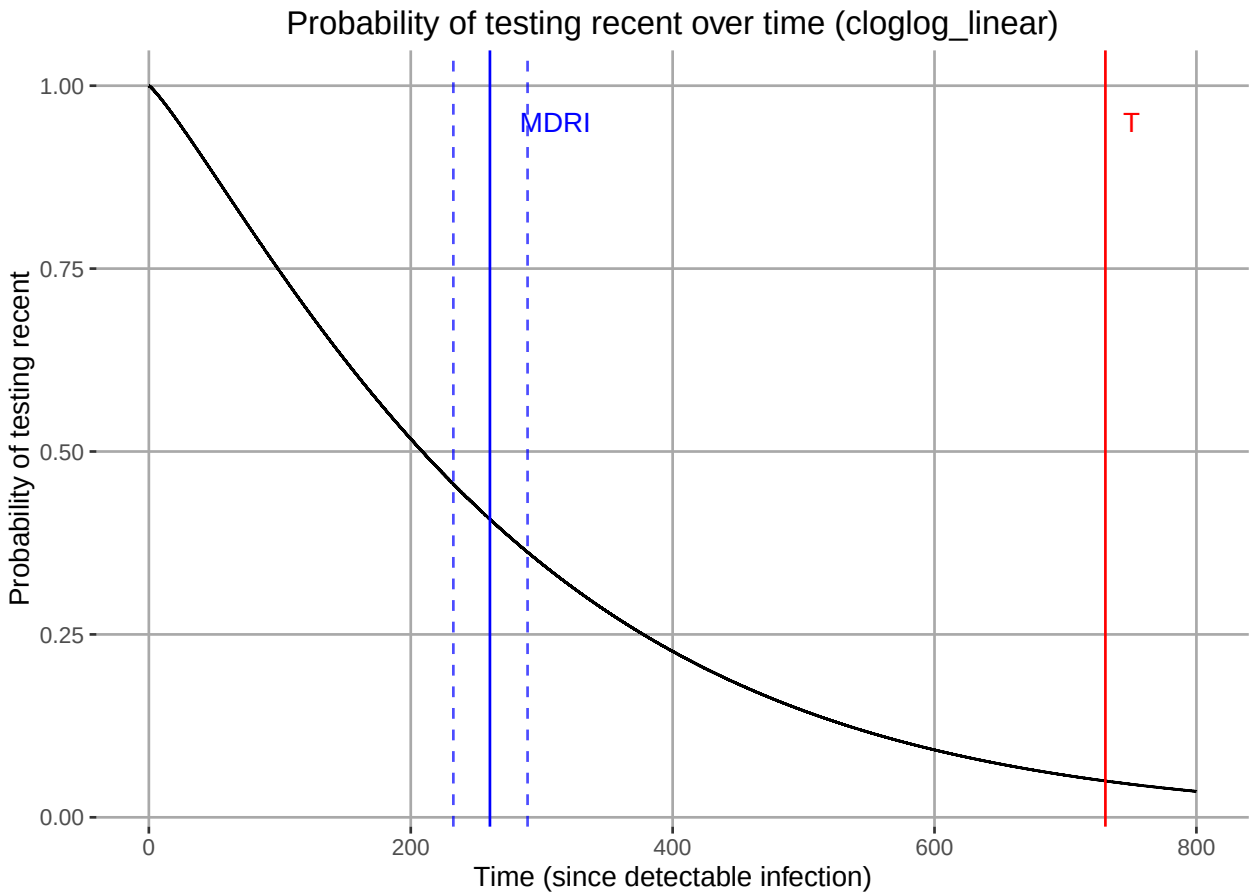
```
mdrical(data=exampledata,
                  subid_var = "SubjectID",
                  time_var = "DaysSinceEDDI",
                  recency_cutoff_time = 730.5,
                  inclusion_time_threshold = 800,
                  functional_forms = c("logit_cubic","cloglog_linear"),
                  recency_rule = "independent_thresholds",
                  recency_vars = c("Result","VL"),
                  recency_params = c(10,0,1000,1),
                  n_bootstraps = 10000,
                  alpha = 0.05,
                  plot = TRUE,
                  parallel = TRUE,
                  cores=4)
```

```
## Loading required package: foreach
```

```
## Loading required package: doMC

## Loading required package: iterators

## Loading required package: parallel

## $MDRI
##                       PE    CI_LB    CI_UB      SD
## logit_cubic    259.2234 231.4433 288.2659 14.4730
## cloglog_linear 260.4783 232.5321 289.2220 14.3735
##
## $Plots
## $Plots$logit_cubic
```




```
##
## $Plots$cloglog_linear
```




```
##
##
## $Models
## $Models$logit_cubic
##
## Call:  glm2::glm2(formula = recency_status ~ 1 + I(time_since_eddi) +
##     I(time_since_eddi^2) + I(time_since_eddi^3), family = binomial(link = "logit"),
##     data = data, control = glm.control(epsilon = tolerance, maxit = maxit,
##         trace = FALSE))
##
## Coefficients:
##          (Intercept)    I(time_since_eddi)  I(time_since_eddi^2)
##            2.554e+00            -1.591e-02             2.184e-05
## I(time_since_eddi^3)
##           -1.471e-08
##
## Degrees of Freedom: 643 Total (i.e. Null);  640 Residual
## Null Deviance:       875.9
## Residual Deviance: 635.3     AIC: 643.3
##
## $Models$cloglog_linear
##
## Call:  glm2::glm2(formula = (1 - recency_status) ~ 1 + I(log(time_since_eddi)),
##     family = binomial(link = "cloglog"), data = data, control = glm.control(epsilon = tolerance,
##         maxit = maxit, trace = FALSE))
```

```
##
## Coefficients:
##             (Intercept)  I(log(time_since_eddi))
##                  -6.618                   1.170
##
## Degrees of Freedom: 643 Total (i.e. Null);  642 Residual
## Null Deviance:       875.9
## Residual Deviance: 637.5     AIC: 641.5
```

Estimating FRR using binomial proportions with `frrcal`

FRR is simply the binomially estimated probability of a *subject's* measurements post-T being 'recent' on the recency test. A binomial exact test is performed using `binom.test`. All of a subject's measurements post-T are evaluated and if the majority are recent, the subject is considered to have measured falsely recent. Inversely, if a majority are non-recent, the subject contritbutes a 'true recent' result. Each subject represents one trial. In the case that exactly half of a subject's measurements are recent, they contribute 0.5 to the outcomes (which are rounded up to the nearest integer over all subjects).

This example calculates a false-recent rate, treating the data at subject level:

```
frrcal(data=exampledata,
             subid_var = "SubjectID",
             time_var = "DaysSinceEDDI",
             recency_cutoff_time = 730.5,
             recency_rule = "independent_thresholds",
             recency_vars = c("Result","VL"),
             recency_params = c(10,0,1000,1),
             alpha = 0.05)
```

```
##  FRRest     LB     UB alpha n_recent n_subjects n_observations
##  0.0301 0.0131 0.0584  0.05        8        266            732
```