Simulation Study

```r
library(survival)
library(glmnet)
```

```
## Loading required package: Matrix
```

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

```
## Loaded glmnet 2.0-16
```

```r
library(polspline)
library(knitr)
library(EnvStats)
```

```
##
## Attaching package: 'EnvStats'
```

```
## The following object is masked from 'package:Matrix':
##
##     print
```

```
## The following objects are masked from 'package:stats':
##
##     predict, predict.lm
```

```
## The following object is masked from 'package:base':
##
##     print.default
```

```r
library(tictoc)
library(abind)
library(bda)
library(tryCatchLog)
```

```
## futile.logger not found. Using tryCatchLog-internal functions for logging...
```

Function for Simulating Survival Time from a Weibull or Lognormal Distribution

This function is based on simulate_data in https://cran.r-project.org/web/packages/rsimsum/vignettes/relhaz.html

```r
#' Simulate survival times with censoring, based on a user-defined distribution
#'
#' This function simulates survival times with censoring, according to a specific
#' distribution that the user parameterizes. The survival times are
#' simulated for user-given covariates and coefficients, and the censoring times
#' are simulated for a user-given distribution.
#' @param dist distribution for the survival times, expected to be "weibull" or "lnorm"
#' @param x model matrix of covariate values
#' @param fcts_select subset of fcts from a hare object containing the coefficients of interest.
#' @param params parameters scale and shape/sigma for the baseline hazard
#' @param FUN random generation function for the distribution of censoring times,
#' expected to be uniform, exponential, or weibull.
#' @param ... arguments for FUN, the random generation function for the
#' censoring distribution
#' @return dataframe appending survival time and censoring indicator to the model matrix x
#' @export
simulate_dist <- function(dist, x, fcts_select, params, FUN, ...) {

  n <- nrow(x)

  # extract unique list of covariates selected
  cov_nums <- sort(fcts_select[,1][fcts_select[,1] != 0])
  cov_names <- colnames(x)[cov_nums]
  x_select <- x[,cov_names]

  # extract the coefficient values from fcts_select
  betas <- fcts_select[,5][fcts_select[,1] != 0]
  betas <- betas[order(fcts_select[,1][fcts_select[,1] != 0])]

  # simulate survival times
  if (dist == "weibull") {

    # simulate survival times according to Bender et al. (2005)
    u <- runif(n)
    time <- (-log(u) / (params$scale * exp(x_select %*% betas)))^(1 / params$shape)

  } else if (dist == "lnorm") {

    z <- rnorm(n)
    time <- exp(params$scale + x_select %*% betas + params$sigma * z)

  } else {stop("Unrecognized Distribution")}



  # Winsorising tiny values for time (smaller than one day on a yearly-scale, e.g. 1 / 365.242),
  # and adding a tiny amount of white noise not to have too many concurrent values
  time <- ifelse(time < 1 / 365.242, 1 / 365.242, time)
  time[time == 1 / 365.242] <- time[time == 1 / 365.242] +
    rnorm(length(time[time == 1 / 365.242]), mean = 0, sd = 1e-4)
  # ...and make sure that the resulting value is positive
  time <- abs(time)

  # Censoring
  cid_time <- FUN(n, ...)

  cid <- ifelse(time < cid_time, 1, 0)

  time <- pmin(time, cid_time)

  # return a dataframe
  data.frame(time, cid, x)
}
```

Function for Simulating Survival Time from a HARE model

```r
#' Simulate survival times with censoring, based on a HARE model
#'
#' This function simulates survival times with censoring, according to a user-given
#' HARE model. The survival times are simulated for covariates provided by the user
#' and coefficients provided by the HARE model, and the censoring times
#' are simulated for a user-given distribution.
#' @param x model matrix of covariate values
#' @param fit hare object to generate survival times from
#' @param FUN random generation function for the distribution of censoring times,
#' expected to be uniform, exponential, or weibull.
#' @param ... arguments for FUN, the random generation function for the
#' censoring distribution
#' @return dataframe appending survival time and censoring indicator to the model matrix x
#' @export
simulate_hare <- function(x, fit, FUN, ...) {

  n <- nrow(x)

  # simulate survival times
  time <- rhare(1, x, fit)



  # Winsorising tiny values for time (smaller than one day on a yearly-scale, e.g. 1 / 365.242),
  # and adding a tiny amount of white noise not to have too many concurrent values
  time <- ifelse(time < 1 / 365.242, 1 / 365.242, time)
  time[time == 1 / 365.242] <- time[time == 1 / 365.242] +
    rnorm(length(time[time == 1 / 365.242]), mean = 0, sd = 1e-4)
  # ...and make sure that the resulting value is positive
  time <- abs(time)

  # Censoring
  cid_time <- FUN(n, ...)

  cid <- ifelse(time < cid_time, 1, 0)

  time <- pmin(time, cid_time)

  # return a dataframe
  data.frame(time, cid, x)
}
```

Setting up the requisite parameters for simulation

```r
load("actg175.RData")

x <- model.matrix( ~ trt + age + wtkg + hemo + drugs +
                     karnof + oprior + preanti + race +
                     gender + symptom + offtrt + cd40 +
                     cd80, actg175)[,-1]

nphm_hare <- readRDS("nphm_hare.rds")
```

```
# extracting the coefficients for basis functions
# that do not correspond to knots and/or tensor products
fcts <- nphm_hare$fcts
fcts_select <- fcts[fcts[,2] == 0 & is.na(fcts[,3]),]

# Standard Deviation of log(Survival Time), needed for the scale parameters
# of the simulating distributions
sigma <- sd(log(actg175$time))
```

Generating example data set from the Weibull distribution

I calculate the Weibull shape parameter and use an arbitrary Weibull scale parameter to make the survival times and censoring rate be similar to those of the ACTG-175 data set.

```
set.seed(1)

parm_res <- fit.Weibull(rhare(100000, cov = rep(0, nphm_hare$ncov), nphm_hare), dist="Weibull")
```

```
set.seed(2)

sim_mat <- simulate_dist(dist = "weibull", x, fcts_select,
                         params = list(scale = 3.347861e-140,
                                       shape = parm_res$pars[2]),
                         FUN = rexp, .4)

summary(sim_mat$time)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00004 0.67606 1.63962 1.76214 3.12540 3.40846
```

```
# censoring rate for the simulated data set
1 - mean(sim_mat$cid)
```

```
## [1] 0.7400655
```

Generating example data set from the Log-normal distribution

I use an arbitrary log-normal scale parameter and a calculated sigma parameter to make the survival times and censoring rate be similar to those of the ACTG-175 data set.

```
set.seed(2)

sim_mat_lnorm <- simulate_dist(dist = "lnorm", x, fcts_select,
                               params = list(scale = -11,
                                             sigma = sigma),
                               FUN = rexp, 2.1)

summary(sim_mat_lnorm$time)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
## 0.0000645 0.0569160 0.1846256 0.3406310 0.4660036 3.1566504
```

```
# censoring rate for the simulated data set
1 - mean(sim_mat_lnorm$cid)
```

```
## [1] 0.7288453
```

Generating example data set from the HARE model

```
set.seed(2)

sim_mat_hare <- simulate_hare(x, nphm_hare,
                              FUN = runif, 0, 3.3)

summary(sim_mat_hare$time)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
## 0.002189 0.732549 1.355970 1.422104 2.033445 3.295673
```

```
# censoring rate for the simulated data set
1 - mean(sim_mat_hare$cid)
```

```
## [1] 0.7358579
```

Coxph Simulation

Weibull distributed survival times

```
phm_sim_mat <- coxph(Surv(time, cid) ~ ., data = sim_mat)

summary(phm_sim_mat)
```

```
## Call:
## coxph(formula = Surv(time, cid) ~ ., data = sim_mat)
## 
##   n= 2139, number of events= 556 
## 
##                  coef  exp(coef)   se(coef)      z Pr(>|z|)    
## trtZDV.ddi -6.803e-01  5.065e-01  1.271e-01 -5.352 8.68e-08 ***
## trtZDV.ZAL -4.565e-01  6.335e-01  1.284e-01 -3.556 0.000376 ***
## trtddi     -4.939e-01  6.103e-01  1.237e-01 -3.992 6.55e-05 ***
## age        -1.212e-02  9.880e-01  5.561e-03 -2.180 0.029282 *  
## wtkg       -2.314e-03  9.977e-01  3.275e-03 -0.707 0.479837    
## hemo1      -2.068e-01  8.132e-01  1.690e-01 -1.224 0.221099    
## drugs1     -1.629e-01  8.497e-01  1.371e-01 -1.189 0.234634    
## karnof      1.866e-03  1.002e+00  8.117e-03  0.230 0.818147    
## oprior1    -4.414e-02  9.568e-01  2.830e-01 -0.156 0.876049    
## preanti     5.042e-04  1.001e+00  1.006e-04  5.010 5.43e-07 ***
## race1      -1.926e-02  9.809e-01  1.043e-01 -0.185 0.853435    
## gender1     9.918e-03  1.010e+00  1.395e-01  0.071 0.943329    
## symptom1    4.581e-01  1.581e+00  1.145e-01  4.002 6.28e-05 ***
## offtrt1    -8.817e-01  4.141e-01  1.015e-01 -8.686  < 2e-16 ***
## cd40        3.779e-02  1.039e+00  1.359e-03 27.803  < 2e-16 ***
## cd80        4.250e-04  1.000e+00  9.724e-05  4.370 1.24e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##            exp(coef) exp(-coef) lower .95 upper .95
## trtZDV.ddi    0.5065     1.9745    0.3948    0.6497
## trtZDV.ZAL    0.6335     1.5786    0.4926    0.8147
## trtddi        0.6103     1.6386    0.4789    0.7777
## age           0.9880     1.0122    0.9772    0.9988
## wtkg          0.9977     1.0023    0.9913    1.0041
## hemo1         0.8132     1.2297    0.5839    1.1325
## drugs1        0.8497     1.1769    0.6495    1.1115
## karnof        1.0019     0.9981    0.9861    1.0179
## oprior1       0.9568     1.0451    0.5495    1.6660
## preanti       1.0005     0.9995    1.0003    1.0007
## race1         0.9809     1.0195    0.7996    1.2034
## gender1       1.0100     0.9901    0.7683    1.3276
## symptom1      1.5810     0.6325    1.2633    1.9786
## offtrt1       0.4141     2.4149    0.3394    0.5052
## cd40          1.0385     0.9629    1.0358    1.0413
## cd80          1.0004     0.9996    1.0002    1.0006
##
## Concordance= 0.914  (se = 0.005 )
## Rsquare= 0.512   (max possible= 0.938 )
## Likelihood ratio test= 1536  on 16 df,   p=<2e-16
## Wald test            = 786.3  on 16 df,   p=<2e-16
## Score (logrank) test = 1130  on 16 df,   p=<2e-16
```

```r
cox.zph(phm_sim_mat)
```

```
##                 rho    chisq      p
## trtZDV.ddi  0.03319 6.12e-01 0.4342
## trtZDV.ZAL -0.03747 7.89e-01 0.3743
## trtddi      0.00426 1.01e-02 0.9199
## age         0.01451 1.13e-01 0.7365
## wtkg        0.08006 3.49e+00 0.0618
## hemo1      -0.05042 1.55e+00 0.2136
## drugs1     -0.00714 3.14e-02 0.8594
## karnof     -0.05259 1.72e+00 0.1894
## oprior1    -0.00073 2.95e-04 0.9863
## preanti    -0.00977 5.67e-02 0.8118
## race1       0.06856 2.70e+00 0.1001
## gender1    -0.00179 1.84e-03 0.9658
## symptom1   -0.02961 4.87e-01 0.4852
## offtrt1     0.03329 6.27e-01 0.4284
## cd40        0.00461 1.23e-02 0.9119
## cd80        0.04935 1.54e+00 0.2143
## GLOBAL           NA 2.04e+01 0.2046
```

Log-normal distributed survival times

```r
phm_sim_mat_lnorm <- coxph(Surv(time, cid) ~ ., data = sim_mat_lnorm)

summary(phm_sim_mat_lnorm)
```

```
## Call:
## coxph(formula = Surv(time, cid) ~ ., data = sim_mat_lnorm)
## 
##   n= 2139, number of events= 580 
## 
##                  coef  exp(coef)   se(coef)       z Pr(>|z|)    
## trtZDV.ddi  0.9188650  2.5064439  0.1367729   6.718 1.84e-11 ***
## trtZDV.ZAL  1.0654428  2.9021238  0.1265975   8.416  < 2e-16 ***
## trtddi      0.8143017  2.2575986  0.1285474   6.335 2.38e-10 ***
## age         0.0105286  1.0105842  0.0055209   1.907  0.05651 .  
## wtkg       -0.0002270  0.9997730  0.0031204  -0.073  0.94200    
## hemo1       0.1042017  1.1098243  0.1620453   0.643  0.52020    
## drugs1      0.1467529  1.1580677  0.1353055   1.085  0.27810    
## karnof     -0.0159226  0.9842035  0.0068964  -2.309  0.02095 *  
## oprior1     0.4690807  1.5985240  0.2392082   1.961  0.04988 *  
## preanti    -0.0011414  0.9988593  0.0001103 -10.348  < 2e-16 ***
## race1      -0.3257693  0.7219717  0.1026675  -3.173  0.00151 ** 
## gender1     0.2151600  1.2400602  0.1313460   1.638  0.10140    
## symptom1   -0.3039862  0.7378710  0.1053642  -2.885  0.00391 ** 
## offtrt1     1.2134183  3.3649675  0.0953360  12.728  < 2e-16 ***
## cd40       -0.0546852  0.9467832  0.0017639 -31.003  < 2e-16 ***
## cd80       -0.0009773  0.9990232  0.0001085  -9.006  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
##            exp(coef) exp(-coef) lower .95 upper .95
## trtZDV.ddi    2.5064     0.3990    1.9171    3.2770
## trtZDV.ZAL    2.9021     0.3446    2.2644    3.7194
## trtddi        2.2576     0.4429    1.7548    2.9045
## age           1.0106     0.9895    0.9997    1.0216
## wtkg          0.9998     1.0002    0.9937    1.0059
## hemo1         1.1098     0.9010    0.8078    1.5247
## drugs1        1.1581     0.8635    0.8883    1.5098
## karnof        0.9842     1.0161    0.9710    0.9976
## oprior1       1.5985     0.6256    1.0002    2.5547
## preanti       0.9989     1.0011    0.9986    0.9991
## race1         0.7220     1.3851    0.5904    0.8829
## gender1       1.2401     0.8064    0.9586    1.6041
## symptom1      0.7379     1.3553    0.6002    0.9071
## offtrt1       3.3650     0.2972    2.7915    4.0563
## cd40          0.9468     1.0562    0.9435    0.9501
## cd80          0.9990     1.0010    0.9988    0.9992
## 
## Concordance= 0.979  (se = 0.001 )
## Rsquare= 0.738   (max possible= 0.98 )
## Likelihood ratio test= 2864  on 16 df,   p=<2e-16
## Wald test            = 1010  on 16 df,   p=<2e-16
## Score (logrank) test = 1335  on 16 df,   p=<2e-16
```

```r
cox.zph(phm_sim_mat_lnorm)
```

```
##                 rho    chisq        p
## trtZDV.ddi  0.15089  19.2296 1.16e-05
## trtZDV.ZAL  0.09262   5.0994 2.39e-02
## trtddi      0.07147   3.1417 7.63e-02
## age         0.19228  27.8444 1.31e-07
## wtkg        0.03185   0.6032 4.37e-01
## hemo1      -0.03097   0.5708 4.50e-01
## drugs1     -0.06752   3.1470 7.61e-02
## karnof      0.12501  10.1582 1.44e-03
## oprior1    -0.02810   0.4754 4.91e-01
## preanti    -0.01580   0.1802 6.71e-01
## race1       0.03004   0.5869 4.44e-01
## gender1    -0.00789   0.0451 8.32e-01
## symptom1   -0.14284  13.9136 1.91e-04
## offtrt1     0.22695  41.8670 9.77e-11
## cd40       -0.35096 159.0881 1.79e-36
## cd80       -0.12101   8.7352 3.12e-03
## GLOBAL           NA 247.0840 2.05e-43
```

Survival times from the HARE non-PH model

```r
phm_sim_mat_hare <- coxph(Surv(time, cid) ~ ., data = sim_mat_hare)

summary(phm_sim_mat_hare)
```

```
## Call:
## coxph(formula = Surv(time, cid) ~ ., data = sim_mat_hare)
##
##   n= 2139, number of events= 565
##
##                  coef  exp(coef)   se(coef)       z Pr(>|z|)
## trtZDV.ddi -3.499e+00  3.023e-02  1.814e-01 -19.286  < 2e-16 ***
## trtZDV.ZAL -2.602e+00  7.416e-02  1.353e-01 -19.226  < 2e-16 ***
## trtddi     -1.937e+00  1.442e-01  1.196e-01 -16.190  < 2e-16 ***
## age         1.815e-03  1.002e+00  5.991e-03   0.303  0.76189
## wtkg        8.075e-03  1.008e+00  3.310e-03   2.440  0.01470 *
## hemo1       9.274e-02  1.097e+00  1.681e-01   0.552  0.58115
## drugs1      3.992e-01  1.491e+00  1.301e-01   3.069  0.00215 **
## karnof     -2.140e-02  9.788e-01  6.716e-03  -3.187  0.00144 **
## oprior1    -1.079e-01  8.977e-01  2.657e-01  -0.406  0.68474
## preanti     1.408e-03  1.001e+00  9.860e-05  14.281  < 2e-16 ***
## race1      -2.850e-01  7.520e-01  1.146e-01  -2.487  0.01289 *
## gender1     9.150e-01  2.497e+00  1.724e-01   5.308 1.11e-07 ***
## symptom1    1.774e+00  5.892e+00  1.059e-01  16.740  < 2e-16 ***
## offtrt1     2.554e+00  1.286e+01  1.093e-01  23.358  < 2e-16 ***
## cd40       -1.120e-02  9.889e-01  5.037e-04 -22.241  < 2e-16 ***
## cd80        1.523e-03  1.002e+00  9.425e-05  16.155  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##            exp(coef) exp(-coef) lower .95 upper .95
## trtZDV.ddi   0.03023   33.08264   0.02118   0.04314
## trtZDV.ZAL   0.07416   13.48454   0.05688   0.09668
## trtddi       0.14420    6.93492   0.11406   0.18230
## age          1.00182    0.99819   0.99012   1.01365
## wtkg         1.00811    0.99196   1.00159   1.01467
## hemo1        1.09717    0.91143   0.78922   1.52530
## drugs1       1.49065    0.67085   1.15516   1.92358
## karnof       0.97883    1.02163   0.96603   0.99180
## oprior1      0.89773    1.11392   0.53328   1.51125
## preanti      1.00141    0.99859   1.00122   1.00160
## race1        0.75202    1.32976   0.60073   0.94140
## gender1      2.49677    0.40052   1.78089   3.50042
## symptom1     5.89195    0.16972   4.78713   7.25175
## offtrt1     12.85844    0.07777  10.37806  15.93163
## cd40         0.98886    1.01126   0.98789   0.98984
## cd80         1.00152    0.99848   1.00134   1.00171
##
## Concordance= 0.93  (se = 0.004 )
## Rsquare= 0.544   (max possible= 0.97 )
## Likelihood ratio test= 1681  on 16 df,   p=<2e-16
## Wald test            = 1377  on 16 df,   p=<2e-16
## Score (logrank) test = 1533  on 16 df,   p=<2e-16
```

```r
cox.zph(phm_sim_mat_hare)
```

```
##                 rho    chisq        p
## trtZDV.ddi -0.01153 1.18e-01 7.31e-01
## trtZDV.ZAL -0.09367 5.70e+00 1.69e-02
## trtddi     -0.04545 1.22e+00 2.69e-01
## age        -0.24583 6.98e+01 6.50e-17
## wtkg        0.03191 5.89e-01 4.43e-01
## hemo1      -0.05443 1.88e+00 1.70e-01
## drugs1      0.06098 2.28e+00 1.31e-01
## karnof      0.00145 1.21e-03 9.72e-01
## oprior1     0.08569 4.28e+00 3.85e-02
## preanti     0.01968 3.23e-01 5.70e-01
## race1       0.02890 6.25e-01 4.29e-01
## gender1    -0.05708 3.48e+00 6.20e-02
## symptom1    0.06746 2.79e+00 9.47e-02
## offtrt1    -0.28861 5.26e+01 4.13e-13
## cd40        0.02803 4.89e-01 4.84e-01
## cd80        0.05120 1.79e+00 1.81e-01
## GLOBAL           NA 1.84e+02 1.50e-30
```

Weibull model (to compare with Coxph), (Weibull distributed survival times)

Weibull Distributed Survival Time

The Weibull model should be more powerful (with less variance in the coefficient estimates) than the Cox Proportional Hazards model, as the data is from a Weibull distribution. The coefficient estimates themselves should be similar.

```r
summary(survreg(Surv(time, cid) ~ ., data = sim_mat))
```

```
## Warning in survreg.fit(X, Y, weights, offset, init = init, controlvals =
## control, : Ran out of iterations and did not converge
```

```
##
## Call:
## survreg(formula = Surv(time, cid) ~ ., data = sim_mat)
##                 Value Std. Error    z      p
## (Intercept) -1.70e-01   0.00e+00 -Inf <2e-16
## trtZDV.ddi   2.34e-03   0.00e+00  Inf <2e-16
## trtZDV.ZAL  -3.09e-02   0.00e+00 -Inf <2e-16
## trtddi       7.25e-02   0.00e+00  Inf <2e-16
## age         -2.04e-03   0.00e+00 -Inf <2e-16
## wtkg         2.04e-03   0.00e+00  Inf <2e-16
## hemo1        5.59e-03   0.00e+00  Inf <2e-16
## drugs1       4.08e-02   0.00e+00  Inf <2e-16
## karnof       2.08e-03   0.00e+00  Inf <2e-16
## oprior1      1.07e-01   0.00e+00  Inf <2e-16
## preanti      5.89e-05   0.00e+00  Inf <2e-16
## race1       -2.10e-02   0.00e+00 -Inf <2e-16
## gender1      5.06e-02   0.00e+00  Inf <2e-16
## symptom1     7.40e-02   0.00e+00  Inf <2e-16
## offtrt1     -3.47e-02   0.00e+00 -Inf <2e-16
## cd40        -2.03e-04   0.00e+00 -Inf <2e-16
## cd80         5.13e-05   0.00e+00  Inf <2e-16
## Log(scale)  -4.08e+00   0.00e+00 -Inf <2e-16
##
## Scale= 0.017
##
## Weibull distribution
## Loglik(model)= -253954   Loglik(intercept only)= 752.9
##  Chisq= -509413.8 on 16 degrees of freedom, p= 1
## Number of Newton-Raphson Iterations: 30
## n= 2139
```

Has an error: "Ran out of iterations and did not converge."

Log-normal Distributed Survival Time

```r
summary(survreg(Surv(time, cid) ~ ., data = sim_mat_lnorm))
```

```
##
## Call:
## survreg(formula = Surv(time, cid) ~ ., data = sim_mat_lnorm)
##                 Value Std. Error      z       p
## (Intercept) -1.08e+01   5.73e-01 -18.91 < 2e-16
## trtZDV.ddi  -2.58e-01   1.02e-01  -2.53 0.01149
## trtZDV.ZAL  -6.41e-01   9.12e-02  -7.02 2.2e-12
## trtddi      -4.08e-01   9.20e-02  -4.44 9.1e-06
## age         -8.33e-03   3.78e-03  -2.20 0.02759
## wtkg         1.18e-03   2.43e-03   0.48 0.62785
## hemo1        3.43e-03   1.21e-01   0.03 0.97750
## drugs1       3.93e-01   1.07e-01   3.69 0.00023
## karnof       2.24e-02   5.06e-03   4.42 9.9e-06
## oprior1     -2.25e-01   1.78e-01  -1.26 0.20656
## preanti      6.34e-04   7.57e-05   8.38 < 2e-16
## race1        1.38e-01   7.51e-02   1.84 0.06595
## gender1     -4.52e-02   9.52e-02  -0.47 0.63489
## symptom1    -5.07e-02   7.62e-02  -0.67 0.50600
## offtrt1     -6.26e-01   6.71e-02  -9.33 < 2e-16
## cd40         2.77e-02   4.80e-04  57.70 < 2e-16
## cd80         7.05e-04   8.32e-05   8.47 < 2e-16
## Log(scale)  -2.97e-01   2.94e-02 -10.09 < 2e-16
##
## Scale= 0.743
##
## Weibull distribution
## Loglik(model)= 926   Loglik(intercept only)= -389.6
##  Chisq= 2631.29 on 16 degrees of freedom, p= 0
## Number of Newton-Raphson Iterations: 10
## n= 2139
```

For some reason Survreg works with the log-normal distributed survival times but not the Weibull-distributed survival times, ironically enough.

Survival times from the HARE non-PH model

```
summary(survreg(Surv(time, cid) ~ ., data = sim_mat_hare))
```

```
##
## Call:
## survreg(formula = Surv(time, cid) ~ ., data = sim_mat_hare)
##                 Value Std. Error      z       p
## (Intercept)  3.88e-01   1.45e-01   2.68  0.0074
## trtZDV.ddi   6.61e-01   2.87e-02  23.01 < 2e-16
## trtZDV.ZAL   4.91e-01   2.21e-02  22.26 < 2e-16
## trtddi       3.66e-01   2.05e-02  17.90 < 2e-16
## age          2.38e-05   1.11e-03   0.02  0.9830
## wtkg        -1.57e-03   6.12e-04  -2.57  0.0102
## hemo1       -1.38e-02   3.13e-02  -0.44  0.6591
## drugs1      -7.66e-02   2.38e-02  -3.22  0.0013
## karnof       4.06e-03   1.24e-03   3.28  0.0010
## oprior1      1.61e-02   4.94e-02   0.33  0.7446
## preanti     -2.63e-04   1.74e-05 -15.16 < 2e-16
## race1        5.53e-02   2.12e-02   2.60  0.0092
## gender1     -1.77e-01   3.16e-02  -5.59 2.3e-08
## symptom1    -3.33e-01   1.81e-02 -18.43 < 2e-16
## offtrt1     -4.77e-01   1.84e-02 -25.92 < 2e-16
## cd40         2.09e-03   8.97e-05  23.33 < 2e-16
## cd80        -2.85e-04   1.60e-05 -17.76 < 2e-16
## Log(scale)  -1.68e+00   2.96e-02 -56.76 < 2e-16
##
## Scale= 0.186
##
## Weibull distribution
## Loglik(model)= -372.8   Loglik(intercept only)= -1239.6
##  Chisq= 1733.61 on 16 degrees of freedom, p= 0
## Number of Newton-Raphson Iterations: 8
## n= 2139
```

Glmnet Simulation

Weibull distributed survival times

```
cv_phmnet <- cv.glmnet(as.matrix(sim_mat[-c(1,2)]),
                       Surv(sim_mat$time, sim_mat$cid),
                       family = "cox", alpha = .95)
```

```
coef(cv_phmnet, s = cv_phmnet$lambda.1se)
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## trtZDV.ddi -0.1165921374
## trtZDV.ZAL  .
## trtddi      .
## age        -0.0006033230
## wtkg        .
## hemo1       .
## drugs1      .
## karnof      .
## oprior1     .
## preanti     0.0002105510
## race1       .
## gender1     .
## symptom1    0.1497432106
## offtrt1    -0.5613652945
## cd40        0.0292327114
## cd80        0.0002023108
```

Log-normal distributed survival times

```
cv_phmnet_lnorm <- cv.glmnet(as.matrix(sim_mat_lnorm[-c(1,2)]),
                             Surv(sim_mat_lnorm$time, sim_mat_lnorm$cid),
                             family = "cox", alpha = .95)
```

```
coef(cv_phmnet_lnorm, s = cv_phmnet_lnorm$lambda.1se)
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## trtZDV.ddi  0.2460842728
## trtZDV.ZAL  0.4135377815
## trtddi      0.1388931068
## age         0.0014953681
## wtkg        .
## hemo1       .
## drugs1      .
## karnof     -0.0041914102
## oprior1     0.0608399620
## preanti    -0.0006390870
## race1      -0.1375975867
## gender1     .
## symptom1   -0.0763084223
## offtrt1     0.7908126506
## cd40       -0.0396867126
## cd80       -0.0006388184
```

Survival times from the HARE non-PH model

```
cv_phmnet_hare <- cv.glmnet(as.matrix(sim_mat_hare[-c(1,2)]),
                            Surv(sim_mat_hare$time, sim_mat_hare$cid),
                            family = "cox", alpha = .95)
```

```
coef(cv_phmnet_hare, s = cv_phmnet_hare$lambda.1se)
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                         1
## trtZDV.ddi -1.499532e-02
## trtZDV.ZAL  .
## trtddi      .
## age         .
## wtkg        .
## hemo1       .
## drugs1      .
## karnof      .
## oprior1     .
## preanti     3.585725e-04
## race1       .
## gender1     .
## symptom1    5.219027e-01
## offtrt1     9.654015e-01
## cd40       -4.301906e-03
## cd80        8.683353e-05
```

PH HARE Simulation

Weibull distributed survival times

```
phm_hare <- hare(sim_mat$time, sim_mat$cid, as.matrix(sim_mat[-c(1,2)]), prophaz = TRUE)

phm_hare$fcts
```

```
##   dim1 knot1 dim2 knot2          beta           SE
## 1    0     0   NA    NA  4.810484e+00 1.697890e-01
## 2    0     1   NA    NA -7.798423e+01 2.609708e+00
## 3   15     0   NA    NA  3.708781e-02 1.294960e-03
## 4   14     0   NA    NA -9.184653e-01 9.630566e-02
## 5   10     0   NA    NA  4.645391e-04 9.555504e-05
## 6   13     0   NA    NA  4.330778e-01 1.114339e-01
## 7    16     0   NA    NA  3.559056e-04 9.396533e-05
## 8     1     0   NA    NA -6.494517e-01 1.247080e-01
## 9     3     0   NA    NA -5.211305e-01 1.221323e-01
## 10    2     0   NA    NA -4.659746e-01 1.246135e-01
```

Log-normal distributed survival times

```
phm_hare_lnorm <- hare(sim_mat_lnorm$time, sim_mat_lnorm$cid,
                       as.matrix(sim_mat_lnorm[-c(1,2)]), prophaz = TRUE)

phm_hare_lnorm$fcts
```

```
##    dim1 knot1 dim2 knot2          beta           SE
## 1     0     0   NA    NA  9.376086e+00 7.083705e-01
## 2    15     0   NA    NA -8.013158e-03 3.409217e-03
## 3    15     1   NA    NA -5.987225e-02 4.159880e-03
## 4    14     0   NA    NA  2.798081e+00 3.670815e-01
## 5     0     3   NA    NA -1.320843e+00 2.983359e-01
## 6     0     1   NA    NA -6.959544e+03 1.107726e+03
## 7     0     2   NA    NA -1.666111e+01 1.655523e+00
## 8     1     0   NA    NA  1.318860e+00 1.287100e-01
## 9     2     0   NA    NA  1.138402e+00 1.266781e-01
## 10    3     0   NA    NA  8.543355e-01 1.269481e-01
## 11   13     0   NA    NA -5.797461e-01 1.048808e-01
## 12    4     0   NA    NA  3.908264e-02 7.251078e-03
## 13    4     0   14     0 -3.560878e-02 9.715485e-03
```

Survival times from the HARE non-PH model

```
phm_hare_hare <- hare(sim_mat_hare$time, sim_mat_hare$cid,
                      as.matrix(sim_mat_hare[-c(1,2)]), prophaz = TRUE)

phm_hare_hare$fcts
```

```
##    dim1 knot1 dim2 knot2          beta           SE
## 1     0     0   NA    NA  -3.907479519 5.9204711771
## 2     0     4   NA    NA -11.885202701 0.3747901158
## 3    14     0   NA    NA -59.922762422 7.9367627205
## 4    15     0   NA    NA   0.176834040 0.0472482389
## 5    13     0   NA    NA   6.156215823 0.1996487800
## 6    16     0   NA    NA   0.005426332 0.0001930217
## 7    15     1   NA    NA   0.259120670 0.0683688165
## 8    10     0   NA    NA   0.005253101 0.0001889529
## 9    15     2   NA    NA   0.144113711 0.0060671439
## 10    0     3   NA    NA -11.273696653 0.6286545088
## 11    1     0   NA    NA -10.858985262 0.3367364096
## 12    2     0   NA    NA  -9.923110125 0.3091477299
## 13    3     0   NA    NA  -7.522070391 0.2447183385
## 14    4     0   NA    NA  -0.229942495 0.0120812261
## 15    4     0   14     0   0.547745584 0.0194567998
## 16    4     1   NA    NA  -0.294862357 0.0698541201
## 17    0     2   NA    NA -19.997649406 2.1267722514
## 18   14     0   15     1  -0.511364971 0.0641682861
## 19    4     2   NA    NA   0.981403528 0.0547747230
## 20   14     0   15     0   0.441811411 0.0623140152
## 21    0     1   NA    NA -41.703994044 3.7809773919
## 22   14     0   15     2   0.057269939 0.0062271235
## 23   15     3   NA    NA  -0.615961524 0.0418373595
## 24   14     0   16     0   0.001992000 0.0001857906
## 25   10     0   14     0   0.002012697 0.0001861793
```

non-PH HARE Simulation

Weibull distributed survival times

```
nphm_hare_sim <- hare(sim_mat$time, sim_mat$cid, as.matrix(sim_mat[-c(1,2)]))

(fcts <- nphm_hare_sim$fcts)
```

```
##    dim1 knot1 dim2 knot2          beta           SE
## 1     0     0   NA    NA  4.810484e+00 1.697890e-01
## 2     0     1   NA    NA -7.798423e+01 2.609708e+00
## 3    15     0   NA    NA  3.708781e-02 1.294960e-03
## 4    14     0   NA    NA -9.184653e-01 9.630566e-02
## 5    10     0   NA    NA  4.645391e-04 9.555504e-05
## 6    13     0   NA    NA  4.330778e-01 1.114339e-01
## 7    16     0   NA    NA  3.559056e-04 9.396533e-05
## 8     1     0   NA    NA -6.494515e-01 1.247080e-01
## 9     3     0   NA    NA -5.211303e-01 1.221323e-01
## 10    2     0   NA    NA -4.659744e-01 1.246135e-01
```

```
# Which covariates have an interaction with time (or a knot of it)?
unique(fcts[fcts[,1] == 0 & !is.na(fcts[,3]), 3])
```

```
## numeric(0)
```

The results here seem to be exactly the same as those for the PH HARE model

Log-normal distributed survival times

```
nphm_hare_sim_lnorm <- hare(sim_mat_lnorm$time, sim_mat_lnorm$cid, as.matrix(sim_mat_lnorm[-c(1,2)]))

(fcts <- nphm_hare_sim_lnorm$fcts)
```

```
##    dim1 knot1 dim2 knot2          beta           SE
## 1     0     0   NA    NA  4.061998e+00 1.010650e+00
## 2    15     0   NA    NA  2.022812e-02 5.830756e-03
## 3    15     1   NA    NA -7.912540e-02 5.534135e-03
## 4    14     0   NA    NA  1.482998e+00 9.121504e-02
## 5     0     2   NA    NA  1.006587e+02 1.887440e+01
## 6     0     1   NA    NA -7.807387e+03 1.174774e+03
## 7     0     2   15     0 -6.238626e-01 9.628516e-02
## 8    13     0   NA    NA -5.308365e-01 1.042583e-01
## 9     1     0   NA    NA  1.196092e+00 1.257502e-01
## 10    2     0   NA    NA  1.157569e+00 1.263454e-01
## 11    3     0   NA    NA  8.161224e-01 1.263709e-01
## 12    0     3   NA    NA -3.843820e+00 7.976292e-01
```

```
# Which covariates have an interaction with time (or a knot of it)?
unique(fcts[fcts[,1] == 0 & !is.na(fcts[,3]), 3])
```

```
## [1] 15
```

Survival times from the HARE non-PH model

```
nphm_hare_sim_hare <- hare(sim_mat_hare$time, sim_mat_hare$cid, as.matrix(sim_mat_hare[-c(1,2)]))

(fcts <- nphm_hare_sim_hare$fcts)
```

```
##    dim1 knot1 dim2 knot2          beta           SE
## 1     0     0   NA    NA -161.71615188 8.2338193474
## 2     0     5   NA    NA  -17.47589738 0.8639886069
## 3    14     0   NA    NA  -32.13312588 1.0480971047
## 4    15     0   NA    NA    2.02789492 0.0824673219
## 5    13     0   NA    NA   22.11035366 0.6903267801
## 6    16     0   NA    NA    0.02341482 0.0007275577
## 7    15     1   NA    NA   -0.54877488 0.0841913165
## 8    10     0   NA    NA    0.02356442 0.0007327608
## 9    15     2   NA    NA    0.17878668 0.0361789805
## 10    0     3   NA    NA  -24.08634107 1.2135941787
## 11    1     0   NA    NA  -38.20995419 1.1901246004
## 12    2     0   NA    NA  -35.32136333 1.0963604009
## 13    3     0   NA    NA  -27.44474735 0.8555714085
## 14    0     5   14     0   12.03652401 0.5929186973
## 15   15     4   NA    NA    0.50165861 0.0406940392
## 16    4     0   NA    NA   -0.96238687 0.0312416537
## 17    4     1   NA    NA    1.19674383 0.2347940979
## 18    4     0   14     0    1.77346955 0.0554310495
## 19    4     2   NA    NA   -0.89889619 0.2972950869
## 20    0     2   NA    NA  -62.01558499 2.2946613553
## 21    4     3   NA    NA    2.53643988 0.0849006920
## 22    0     1   NA    NA -147.38919470 5.0116241436
## 23   15     3   NA    NA   -2.31825163 0.0803985075
## 24    0     4   NA    NA  -39.43215052 1.3648319035
## 25    0     6   NA    NA  -23.41777805 0.9002471791
## 26    0     4   14     0   23.43482051 1.0840209190
```

```
# Which covariates have an interaction with time (or a knot of it)?
unique(fcts[fcts[,1] == 0 & !is.na(fcts[,3]), 3])
```

```
## [1] 14
```

Simulating N = 100 Times

```
#' Select and calculate coefficient estimates for Cox PH, penalized PH, HARE PH,
#' and HARE non-PH models based on N simulated data sets.
```

```r
#'
#' This function calls the simulation function for the distribution of interest N
#' times and calculates the corresponding regression coefficient estimates for
#' each simulated data set.
#' @param N number of simulated data sets to fit the models to.
#' @param p number of covariates in original dataset
#' @param hare_sim if TRUE, generate survival times from simulate_hare.
#' If FALSE (default), generate survival times from simulate_dist
#' @param ... arguments for simulate_dist or simulate_hare
#' @return p x 5 x N array containing the coefficient estimates for variables
#' selected among p initial variables by 4 models fitted on N data sets,
#' as well as variables that the non-PH HARE model selects as having
#' an interaction with time (violation of proportional hazards)
#' @export
simulate_regression <- function(N, p, hare_sim = FALSE, ...) {

  res <- array(NA, dim = c(p, 5, N))

  for (i in 1:N) {

    if (!hare_sim) {
      sim_mat <- simulate_dist(...)
    }
    else {
      sim_mat <- simulate_hare(...)
    }

    if (i %in% c(124, 126, 169, 300, 322, 466, 497)) {

      sim_mat_test <- sim_mat

      load("penalized_PH124.RData")

      if (identical(sim_mat_test, sim_mat)) {

        print("This sim_mat will lead to an abortion.")

        next

      }

      load("penalized_PH126.RData")

      if (identical(sim_mat_test, sim_mat)) {

        print("This sim_mat will lead to an abortion.")

        next

      }

      load("penalized_PH169.RData")

      if (identical(sim_mat_test, sim_mat)) {
        print("This sim_mat will lead to an abortion.")
        next
      }
      load("penalized_PH300.RData")
      if (identical(sim_mat_test, sim_mat)) {
        print("This sim_mat will lead to an abortion.")
        next
      }
      load("penalized_PH322.RData")
      if (identical(sim_mat_test, sim_mat)) {
        print("This sim_mat will lead to an abortion.")
        next
      }
      load("penalized_PH466.RData")
      if (identical(sim_mat_test, sim_mat)) {
        print("This sim_mat will lead to an abortion.")
        next
      }
      load("penalized_PH497.RData")
      if (identical(sim_mat_test, sim_mat)) {
        print("This sim_mat will lead to an abortion.")
        next
      }
      sim_mat <- sim_mat_test
    }
    tryCatchLog({
      # Cox Proportional Hazards model
      phm_sim_mat <- coxph(Surv(time, cid) ~ ., data = sim_mat)

      s_phm_sim_mat <- summary(phm_sim_mat)

      # only storing the coefficient values for which the p-value is <= .05
      res[s_phm_sim_mat$coefficients[,5] <= .05, 1, i] <-
        phm_sim_mat$coefficients[s_phm_sim_mat$coefficients[,5] <= .05]

    }, error = function(e) {save(sim_mat, file = paste0("error_sim_mats/", "PH", i,".RData"))
      print(paste("error", i, "PH"))
      },
    warning = function(w) {save(sim_mat, file = paste0("warning_sim_mats/", "PH", i,".RData"))
      print(paste("warning", i, "PH"))
      })



    tryCatchLog({

      # Penalized Proportional Hazards model
      cv_phmnet <- cv.glmnet(as.matrix(sim_mat[-c(1,2)]),
                             Surv(sim_mat$time, sim_mat$cid),
                             family = "cox", alpha = .95)

      selected_coef <- as.numeric(coef(cv_phmnet, s = cv_phmnet$lambda.1se))

      res[selected_coef != 0, 2, i] <- selected_coef[selected_coef != 0]

    }, error = function(e) {save(sim_mat, file = paste0("error_sim_mats/", "penalized_PH", i,".RData"))
      print(paste("error", i, "penalized_PH"))
      },
    warning = function(w) {save(sim_mat, file = paste0("warning_sim_mats/", "penalized_PH", i,".RData"))
      print(paste("warning", i, "penalized_PH"))
      })



    tryCatchLog({

      # PH HARE model
      phm_hare <- hare(sim_mat$time, sim_mat$cid,
                       as.matrix(sim_mat[-c(1,2)]), prophaz = TRUE)

      # extracting the coefficients for basis functions
      # that do not correspond to time, knots, and/or tensor products
      fcts <- phm_hare$fcts
      fcts_select <- fcts[fcts[,1] != 0 & fcts[,2] == 0 & is.na(fcts[,3]),]

      res[fcts_select[,1], 3, i] <- fcts_select[,5]

    }, error = function(e) {save(sim_mat, file = paste0("error_sim_mats/", "hare_PH", i,".RData"))
      print(paste("error", i, "hare_PH"))
```

```r
      },
    warning = function(w) {save(sim_mat, file = paste0("warning_sim_mats/", "hare_PH", i,".RData"))
      print(paste("warning", i, "hare_PH"))
      })



    tryCatchLog({

      # non-PH HARE model
      nphm_hare <- hare(sim_mat$time, sim_mat$cid, as.matrix(sim_mat[-c(1,2)]))

      # extracting the coefficients for basis functions
      # that do not correspond to time, knots, and/or tensor products
      nphm_fcts <- nphm_hare$fcts
      nphm_fcts_select <- nphm_fcts[nphm_fcts[,1] != 0 &
                                      nphm_fcts[,2] == 0 & is.na(nphm_fcts[,3]),]

      res[nphm_fcts_select[,1], 4, i] <- nphm_fcts_select[,5]

      # Covariates that have an interaction with time (or a knot of it)?
      covxtime <- unique(nphm_fcts[nphm_fcts[,1] == 0 & !is.na(nphm_fcts[,3]), 3])

      res[covxtime, 5, i] <- 1

    }, error = function(e) {save(sim_mat, file = paste0("error_sim_mats/", "hare_non-PH", i,".RData"))
      print(paste("error", i, "hare_non_PH"))
      },
    warning = function(w) {save(sim_mat, file = paste0("warning_sim_mats/", "hare_non-PH", i,".RData"))
      print(paste("warning", i, "hare_non_PH"))
      })

  }

  return(res)

}
```

Weibull distributed survival times

```r
# set.seed(636)
#
# tic()
# sims <- simulate_regression(N = 500, p = 16, hare_sim = FALSE, "weibull", x,
#                             fcts_select, params = list(scale = 3.347861e-140,
#                                                        shape = parm_res$pars[2]),
#                             FUN = rexp, rate = .4)
#
# save(sims, file = "sims500.RData")
# toc()
```

2880.832 sec elapsed

Proportion of times each covariate is selected across the four models fitted on simulated survival times from the Weibull distribution (this will be Table 5a in the manuscript)

```
load("sims500.RData")

prop_weibull <- matrix(0, nrow = 16, ncol = 5)

for (i in 1:16) {

  for (j in 1:5) {

    prop_weibull[i,j] <- sum(!is.na(sims[i, j,])) / dim(sims)[3]

  }

}

prop_weibull
```

```
##        [,1]  [,2]  [,3]  [,4]  [,5]
##  [1,] 1.000 0.986 0.992 0.992 0.008
##  [2,] 1.000 0.978 0.982 0.980 0.004
##  [3,] 0.992 0.786 0.966 0.962 0.006
##  [4,] 0.938 0.968 0.826 0.826 0.014
##  [5,] 0.046 0.178 0.004 0.004 0.000
##  [6,] 0.030 0.210 0.010 0.010 0.002
##  [7,] 0.072 0.260 0.012 0.012 0.000
##  [8,] 0.056 0.190 0.008 0.008 0.000
##  [9,] 0.044 0.186 0.008 0.008 0.000
## [10,] 1.000 1.000 0.986 0.986 0.006
## [11,] 0.052 0.202 0.010 0.010 0.000
## [12,] 0.076 0.224 0.008 0.008 0.000
## [13,] 0.962 0.972 0.866 0.868 0.000
## [14,] 1.000 1.000 1.000 1.000 0.008
## [15,] 1.000 1.000 1.000 1.000 0.028
## [16,] 1.000 1.000 0.984 0.984 0.014
```

Log-normal distributed survival times

```
# set.seed(636)
#
# tic()
# sims <- simulate_regression(N = 500, p = 16, hare_sim = FALSE, "lnorm", x,
#                             fcts_select, params = list(scale = -11, sigma = sigma),
#                             FUN = rexp, rate = 2.1)
#
# save(sims, file = "sims_lnorm500.RData")
# toc()
```

4112.399 sec elapsed

Proportion of times each covariate is selected across the four models fitted on simulated survival times from the Log-normal distribution (this will be Table 5b in the manuscript)

```
load("sims_lnorm500.RData")

prop_lnorm <- matrix(0, nrow = 16, ncol = 5)

for (i in 1:16) {

  for (j in 1:5) {

    prop_lnorm[i,j] <- sum(!is.na(sims[i, j,])) / dim(sims)[3]

  }

}

prop_lnorm
```

```
##        [,1]  [,2]  [,3]  [,4]  [,5]
##  [1,] 1.000 0.486 0.998 0.998 0.052
##  [2,] 1.000 0.816 0.998 0.998 0.052
##  [3,] 1.000 0.418 0.998 0.998 0.042
##  [4,] 0.862 0.484 0.922 0.922 0.024
##  [5,] 0.058 0.042 0.024 0.024 0.004
##  [6,] 0.086 0.034 0.064 0.066 0.004
##  [7,] 0.442 0.088 0.070 0.064 0.002
##  [8,] 0.624 0.408 0.030 0.030 0.002
##  [9,] 0.054 0.042 0.036 0.034 0.002
## [10,] 1.000 0.936 0.812 0.812 0.044
## [11,] 0.264 0.122 0.040 0.040 0.004
## [12,] 0.108 0.026 0.054 0.054 0.000
## [13,] 0.994 0.602 0.998 0.998 0.050
## [14,] 1.000 1.000 1.000 1.000 0.180
## [15,] 1.000 1.000 1.000 1.000 0.348
## [16,] 1.000 0.994 0.464 0.464 0.024
```

Survival times from the HARE non-PH model

```
# set.seed(3)
#
# tic()
# sims <- simulate_regression(N = 500, p = 16, hare_sim = TRUE, x, nphm_hare,
#                                 FUN = runif, 0, 3.3)
#
# save(sims, file = "sims_hare500.RData")
# toc()
```

5413.029 sec elapsed

Proportion of times each covariate is selected across the four models fitted on simulated survival times from the Log-normal distribution (this will be Table 5b in the manuscript)

```
load("sims_hare500.RData")

prop_hare <- matrix(0, nrow = 16, ncol = 5)
```

```
for (i in 1:16) {

  for (j in 1:5) {

    prop_hare[i,j] <- sum(!is.na(sims[i, j,])) / dim(sims)[3]

  }

}

prop_hare
```

```
##        [,1]  [,2]  [,3]  [,4]  [,5]
##  [1,] 0.690 0.780 0.814 0.814 0.074
##  [2,] 0.690 0.716 0.830 0.830 0.046
##  [3,] 0.692 0.598 0.808 0.816 0.184
##  [4,] 0.622 0.738 0.944 0.946 0.056
##  [5,] 0.168 0.122 0.060 0.066 0.004
##  [6,] 0.114 0.050 0.056 0.068 0.000
##  [7,] 0.178 0.146 0.040 0.046 0.000
##  [8,] 0.518 0.536 0.068 0.072 0.006
##  [9,] 0.026 0.104 0.036 0.030 0.000
## [10,] 0.690 0.872 0.894 0.892 0.174
## [11,] 0.198 0.094 0.048 0.040 0.002
## [12,] 0.266 0.124 0.022 0.014 0.002
## [13,] 0.692 0.870 0.912 0.918 0.096
## [14,] 0.692 0.924 0.918 0.924 0.542
## [15,] 0.692 0.930 0.936 0.942 0.064
## [16,] 0.690 0.774 0.856 0.858 0.082
```