Multistate example from Crowther & Lambert — with multiple timescales

SDCC
http://bendixcarstensen.com/AdvCoh
February 2021
Version 8

Compiled Tuesday 2nd February, 2021, 21:58
from: /home/bendix/teach/AdvCoh/00/examples/bcMS/bcMS.tex

Bendix Carstensen Steno Diabetes Center Copenhagen, Gentofte, Denmark
& Department of Biostatistics, University of Copenhagen
bcar0029@regionh.dk b@bxc.dk
http://BendixCarstensen.com

Contents

1 Introduction

This is a re-do (and extension) of (parts of) the example from the short-titled paper by Crowther & Lambert [1].

The data provided by the authors has been groomed to a slightly modified form and is included in the `Epi` package, where times of relapse (`tor`), metastasis (`tom`) and death (`tod`) are only non-`NA` for those that actually do see the events. In addition, we have the times of exit from the study (`tox`) and the vital status (Alive/Dead) at `tox`, `xst`.

```
> library(Epi)
> library(popEpi)
> data(BrCa)
> str(BrCa)
'data.frame':	2982 obs. of  17 variables:
 $ pid   : int  1264 1150 838 1214 1130 1118 386 1417 927 489 ...
 $ year  : int  1986 1990 1988 1990 1989 1987 1989 1993 1984 1989 ...
 $ age   : int  54 55 34 42 35 50 46 40 36 42 ...
 $ meno  : Factor w/ 2 levels "pre","post": 2 2 1 2 1 2 2 1 1 1 ...
 $ size  : Factor w/ 3 levels "<=20 mm",">20-50 mm",..: 1 2 1 1 1 1 1 1 1 1 ...
 $ grade : Factor w/ 2 levels "2","3": 1 1 1 1 1 1 1 1 1 1 ...
 $ nodes : int  0 0 0 0 0 0 0 0 0 0 ...
 $ pr    : int  1360 763 113 465 82 75 174 0 43 462 ...
 $ pr.tr : num  7.22 6.64 4.74 6.14 4.42 ...
 $ er    : int  149 763 109 79 25 10 56 2 23 75 ...
 $ hormon: Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ chemo : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ tor   : num  NA NA NA NA NA ...
 $ tom   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ tod   : num  NA NA NA NA NA ...
 $ tox   : num  12.97 8.78 9.41 10.47 10.35 ...
 $ xst   : Factor w/ 2 levels "Alive","Dead": 1 1 1 1 1 2 1 1 1 1 ...
```

1.1 Setting up a `Lexis` object for the follow-up

Now we are in a position to set up the survival data as a Lexis object. The age and date of entry are only given as integral years, so in order to make the data credible we add a random number between 0 and 1 to mimic a real age and date at entry. We define the time scale `tfd` (`time from diagnosis`) as time since entry into the study:

```
> set.seed( 1952 )
> Lbc <- Lexis( entry = list( tfd = 0,
+                               A = age  + runif(nrow(BrCa)),
+                               P = year + runif(nrow(BrCa)) ),
+                exit = list( tfd = tox ),
+         exit.status = xst,
+                  id = pid,
+                data = BrCa )
NOTE: entry.status has been set to "Alive" for all.
> summary( Lbc )

Transitions:
     To
From    Alive Dead  Records:  Events: Risk time:  Persons:
  Alive  1710 1272      2982     1272   21270.74      2982
> names( Lbc )

 [1] "tfd"     "A"       "P"       "lex.dur" "lex.Cst" "lex.Xst" "lex.id"  "pid"
 [9] "year"    "age"     "meno"    "size"    "grade"   "nodes"   "pr"      "pr.tr"
[17] "er"      "hormon"  "chemo"   "tor"     "tom"     "tod"     "tox"     "xst"
```

Now we want to cut the follow up at the times of relapse (including metastasis), but keep track of whether a person died with or without relapse, so we set `split.states` to true, and since time since relapse is presumably of interest too we ask for that time scale to be defined as well (using the argument `new.scale`):

```
> Rbc <- cutLexis(Lbc,
+                cut = pmin(Lbc$tor, Lbc$tom, na.rm=TRUE),
+          timescale = "tfd",
+          new.state = "Rel",
+       split.states = TRUE,
+          new.scale = "tfr")
> summary(Rbc, timeScale = TRUE)
Transitions:
     To
From     Alive  Rel Dead Dead(Rel)  Records:  Events: Risk time:  Persons:
  Alive   1269 1518  195         0      2982     1713   17203.80      2982
  Rel        0  441    0      1077      1518     1077    4066.94      1518
  Sum     1269 1959  195      1077      4500     2790   21270.74      2982

Timescales:
  tfd     A     P   tfr
   ""    ""    "" "Rel"
```

From the summary we see that the transitions to death are to different states, depending on whether a relapse had occurred or not (this is the result of `split.states`), this will eventually allow us to assess the cumulative risk of relapse. Moreover `new.scale` ensured that a new time scale, `tfr`, time from relapse has been added to the `Lexis` object — reflected in the `time.since` column of the summary.

We can illustrate the transitions by a plot that gives a convenient overview of transitions:

```
> boxes(Rbc, boxpos = list(x = c(15,15,85,85),
+                          y = c(85,15,85,15)),
+          show.BE = TRUE,
+          scale.R = 100,
+              cex = 1.1)
```

2 Modeling rates

In line with Crowther and Lambert we now model the transition rates. To this end we first split the data in smaller chunks of length 1 month — with some 20,000 PY we would expect to have some 250,000 records:

```
> system.time(
+ Sbc <- splitLexis(Rbc, breaks = seq(0, 100, 1/12), "tfd"))
   user  system elapsed
  3.293   2.046   5.339
> summary(Sbc)
Transitions:
     To
From      Alive   Rel Dead Dead(Rel)  Records:  Events: Risk time:  Persons:
  Alive 206228  1518  195         0    207941     1713   17203.80      2982
  Rel        0 49251    0      1077     50328     1077    4066.94      1518
  Sum   206228 50769  195      1077    258269     2790   21270.74      2982
```

In the `popEpi` package is a similar function with more elegant syntax and which is somewhat faster, particularly for large data sets:

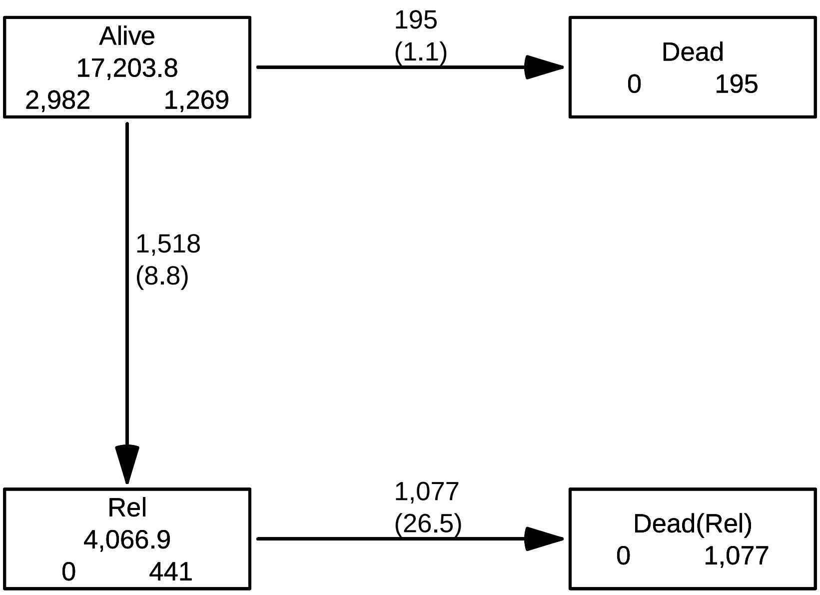

Figure 1: *Transitions in the correctly set up multistate model for the breast cancer survival dataset. Numbers* in *the boxes are person-years and (at the bottom) the number of persons starting resp. ending their follow-up in each state (by the* `show.BE` *argument). Numbers on the arrows are the number of transitions and transition rates per 100 PY (by the* `scale.R` *argument).* ./bcMS-box-r

```
> system.time(
+ Sbc <- splitMulti(Rbc, tfd = seq(0, 100, 1/12)))

   user  system elapsed
  3.186   1.625   4.853

> summary(Sbc)

Transitions:
     To
From      Alive   Rel Dead Dead(Rel)  Records:  Events: Risk time:  Persons:
  Alive 206228  1518  195         0    207941     1713   17203.80      2982
  Rel        0 49251    0      1077     50328     1077    4066.94      1518
  Sum   206228 50769  195      1077    258269     2790   21270.74      2982
```

2.1 Stacking data?

We could model all 3 rates jointly by stacking the data — the function `stack.Lexis` would do this, and create variables `lex.Tr` (transition type) and `lex.Fail` (event indicator):

```
> Stbc <- stack(Sbc)

NOTE: lex.Cst and lex.Xst now have levels:
 Alive Rel Dead Dead(Rel)

> round( ftable(xtabs( cbind('No. records'=lex.id/lex.id,
+                             lex.Fail,
+                             lex.dur) ~ lex.Tr + lex.Xst,
+                       data=Stbc),
+                 row.vars=c(3,1)), 1)
```

```
                          lex.Xst     Alive       Rel      Dead Dead(Rel)
             lex.Tr
No. records  Alive->Rel            206228.0    1518.0     195.0       0.0
             Alive->Dead           206228.0    1518.0     195.0       0.0
             Rel->Dead(Rel)             0.0   49251.0       0.0    1077.0
lex.Fail     Alive->Rel                 0.0    1518.0       0.0       0.0
             Alive->Dead                0.0       0.0     195.0       0.0
             Rel->Dead(Rel)             0.0       0.0       0.0    1077.0
lex.dur      Alive->Rel             17132.8      62.8       8.3       0.0
             Alive->Dead            17132.8      62.8       8.3       0.0
             Rel->Dead(Rel)             0.0    4023.5       0.0      43.5
```

Only `lex.Tr` and `lex.Fail` should be used when modeling rates from stacked data.

However, stacking data is only needed when all transitions are to be modeled jointly, or more specifically, when more than one transition *out* of a given state are modeled jointly. This type of modeling is rarely wanted, since rates of different types of events (in this case relapse and death) are unlikely to depend on the same variables in the same way.

It is much more likely that different mortality rates depend on covariates in the same way — in this case that mortality from "Alive" and from "Rel" depend on time since entry and on the clinical parameters the same way. Additionally we may take time since relapse into account.

In such an instance, the original `Lexis` object where the total follow-up time is represented exactly once in `lex.dur`, will suffice as database for the analysis, because at most *one* transition out of each state is considered. So we shall leave aside the stacking, and model the three rates separately.

2.2 Initial model by C & L

The initial approach is basically to model each of the transitions separately; here we use natural splines with 4 knots placed at the quantiles of the transition times (we refer to the transitions as `ad` (alive to dead), `ar` (alive to relapse), `rd` (relapse to dead). For the sake of completeness we also compute knots on the scale of time since relapse, as well as for the (fixed) difference between `tfd` and `tfr` (the time *at* relapse — note that we do not construct a separate variable for this):

```
> ( kd.ad <- with( subset( Sbc, lex.Cst=="Alive" & lex.Xst=="Dead"),
+                  quantile( tfd+lex.dur, probs=(1:4-0.5)/4) ) )
    12.5%     37.5%     62.5%     87.5%
 1.704312  3.874059  6.058864 10.284052
> ( kd.ar <- with( subset( Sbc, lex.Cst=="Alive" & lex.Xst=="Rel"),
+                  quantile( tfd+lex.dur, probs=(1:4-0.5)/4) ) )
    12.5%     37.5%     62.5%     87.5%
0.8477071 1.8254620 3.3381246 6.8610539
> ( kd.rd <- with( subset( Sbc, lex.Cst=="Rel" & lex.Xst=="Dead(Rel)"),
+                  quantile( tfd+lex.dur, probs=(1:4-0.5)/4) ) )
   12.5%    37.5%    62.5%    87.5%
1.655031 3.091034 5.156742 8.421629
> ( kr.rd <- with( subset( Sbc, lex.Cst=="Rel" & lex.Xst=="Dead(Rel)"),
+                  quantile( tfr+lex.dur, probs=(1:4-0.5)/4) ) )
    12.5%     37.5%     62.5%     87.5%
0.3504449 1.1854894 2.2491443 4.4736482
> ( ka.rd <- with( subset( Sbc, lex.Cst=="Rel" & lex.Xst=="Dead(Rel)"),
+                  quantile( tfd-tfr, probs=(1:4-0.5)/4) ) )
```

```
    12.5%     37.5%     62.5%     87.5%
0.7091033 1.4934976 2.5708419 4.7351130
```

With these vectors of knots in place we can fit models for the three rates — note the similarity of the modeling code for the different models and the immediate readability of what is being modeled; `lex.Cst` is used to define the risk set (using `subset`) and `lex.Xst` to define the event type:

```
> m.ad <- glm(cbind(lex.Xst=="Dead",
+                    lex.dur) ~ Ns(tfd, knots = kd.ad),
+              family = poisreg,
+                data = subset(Sbc, lex.Cst=="Alive"))
> m.ar <- glm(cbind(lex.Xst=="Rel",
+                    lex.dur) ~ Ns(tfd, knots = kd.ar),
+              family = poisreg,
+                data = subset(Sbc, lex.Cst=="Alive"))
> m.rd <- glm(cbind(lex.Xst=="Dead(Rel)",
+                    lex.dur) ~ Ns(tfd, knots = kd.rd),
+              family = poisreg,
+                data = subset(Sbc, lex.Cst=="Rel"))
> x.rd <- update(m.rd, . ~ . + Ns(tfr, knots=kr.rd))
> r.rd <- update(x.rd, . ~ . - Ns(tfd, knots=kd.rd))
> anova( m.rd, x.rd, r.rd, test="Chisq" )
Analysis of Deviance Table

Model 1: cbind(lex.Xst == "Dead(Rel)", lex.dur) ~ Ns(tfd, knots = kd.rd)
Model 2: cbind(lex.Xst == "Dead(Rel)", lex.dur) ~ Ns(tfd, knots = kd.rd) +
    Ns(tfr, knots = kr.rd)
Model 3: cbind(lex.Xst == "Dead(Rel)", lex.dur) ~ Ns(tfr, knots = kr.rd)
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1     50324      10337
2     50321      10260  3   77.541 < 2.2e-16
3     50324      10458 -3 -198.089 < 2.2e-16
```

We see that the mortality rates in relapse depends strongly on the time since relapse, a deviance reduction of 77 on 3 df! Ditching the (non-linear) effect of `tfd` is clearly neither a feasible option with a deviance difference of 198 on 3 df, so there is pretty strong evidence that mortality after relapse depends *both* on time since diagnosis and time since relapse. We shall deal with this extension later.

As an aside, there is a function `glm.Lexis` that exploits the structure of the `Lexis` objects so the model `m.ad` can be fitted by:

```
> M.ad <- glm.Lexis(Sbc, ~ Ns(tfd, knots=kd.ad), to = "Dead")
stats::glm Poisson analysis of Lexis object Sbc with log link:
Rates for the transition: Alive->Dead
> round( cbind( ci.exp(m.ad), ci.exp(M.ad) ), 4 )
                        exp(Est.)   2.5%  97.5% exp(Est.)   2.5%  97.5%
(Intercept)                0.0074 0.0058 0.0094    0.0074 0.0058 0.0094
Ns(tfd, knots = kd.ad)1    1.2194 0.7391 2.0119    1.2194 0.7391 2.0119
Ns(tfd, knots = kd.ad)2    4.1187 2.4189 7.0130    4.1187 2.4189 7.0130
Ns(tfd, knots = kd.ad)3    1.9162 1.3218 2.7780    1.9162 1.3218 2.7780
> attr( M.ad, 'Lexis' )
$data
[1] "Sbc"

$trans
[1] "Alive->Dead"

$formula
```

```
~Ns(tfd, knots = kd.ad)
<environment: 0x47d192e8>

$scale
[1] 1
```

First we turn to the transition rates as function of time since diagnosis. Note that since the `lex.dur` is in units of single person-years, the predicted rates will be in units of events per 1 person-year and so must explicitly scale the predicted rates if we want them in a different unit:

```
> nd <- data.frame(tfd = seq(0, 15, 0.1))
> ad.rate <- ci.pred(m.ad, nd)
> ar.rate <- ci.pred(m.ar, nd)
> rd.rate <- ci.pred(m.rd, nd)
```

We then can plot the three sets of estimated rates in the same graph:

```
> clr <- rainbow(3) ; yl <- c(0.3,200)
> matshade(nd$tfd, cbind(ad.rate,
+                        ar.rate,
+                        rd.rate) * 100, plot = TRUE,
+          type = "l", lty = 1, lwd = 3, col = clr, las = 1,
+          log = "y", xlab = "Time since diagnosis (years)",
+          ylim = yl, ylab = "Rate per 100 PY" )
> text(par("usr")[2]*0.95, (10^par("usr"))[3]*1.4^(1:3),
+      c("Alive -> Dead","Alive -> Relapse","Relapse -> Dead"),
+      col = clr, adj = 1, font = 2 )
> matshade(nd$tfd, ci.ratio(rd.rate, ad.rate),
+          lty = 1, lwd = c(3,1,1), col = gray(0.6))
> abline(h = 1, col = gray(0.6))
```

Note that since the estimates of the transition rates are uncorrelated we can use the `ci.ratio` to derive the mortality rate ratio between persons with and without relapse.

From the graph in figure 2 we see that the occurrence of relapse almost doubles over the first two years and then decays approximately to the initial level at about 5 years. We also observe that the mortality RR between persons with relapse and those without decreases from extremely high (50–100) to about 5, a combination of decreasing mortality among persons with relapse and an increasing mortality among persons without relapse, the latter most likely an effect of age..

3 The two time scales — and their difference

We noted that the model `x.rd` above with effects of both time since diagnosis and time since relapse represented a substantial improvement over the models with only one of these time-scales.

We could expand this model further with an effect of time *at* relapse, `tfd` − `tfr`:

```
>  M.rd <- glm.Lexis(Sbc, ~ Ns(tfd     , knots = kd.rd), to = "Dead(Rel)")

stats::glm Poisson analysis of Lexis object Sbc with log link:
Rates for the transition: Rel->Dead(Rel)

>  X.rd <- glm.Lexis(Sbc, ~ Ns(tfd     , knots = kd.rd) +
+                           Ns(     tfr, knots = kr.rd), to = "Dead(Rel)")

stats::glm Poisson analysis of Lexis object Sbc with log link:
Rates for the transition: Rel->Dead(Rel)
```

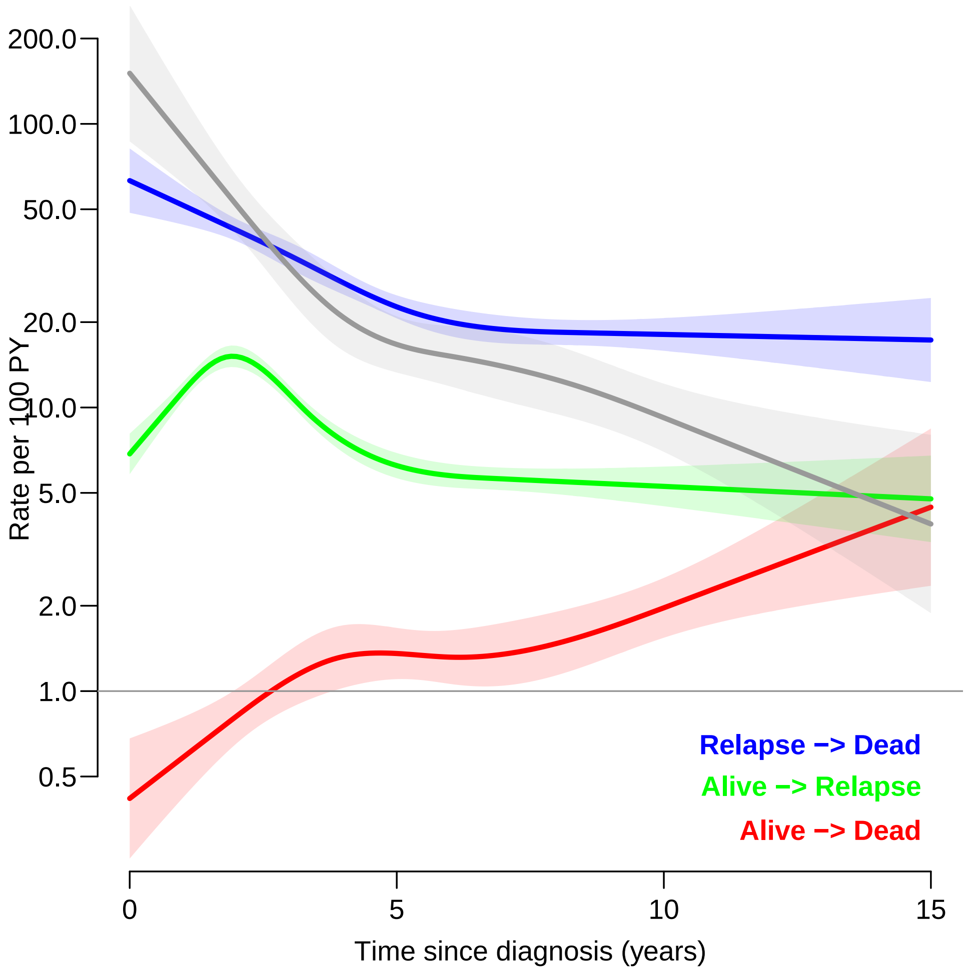

Figure 2: *Transition rates as function of time since diagnosis, the gray line is the mortality rate-ratio between persons with and without relapse — it seems as if the earlier the relapse, the higher the impact on mortality.* `./bcMS-pr-pl`

```
> XX.rd <- glm.Lexis(Sbc, ~ Ns(tfd    , knots = kd.rd) +
+                          Ns(    tfr, knots = kr.rd) +
+                          Ns(tfd-tfr, knots = ka.rd), to = "Dead(Rel)")

stats::glm Poisson analysis of Lexis object Sbc with log link:
Rates for the transition: Rel->Dead(Rel)

> anova( M.rd, X.rd, XX.rd, test = "Chisq" )

Analysis of Deviance Table

Model 1: cbind(trt(Lx$lex.Cst, Lx$lex.Xst) %in% trnam, Lx$lex.dur) ~ Ns(tfd,
    knots = kd.rd)
Model 2: cbind(trt(Lx$lex.Cst, Lx$lex.Xst) %in% trnam, Lx$lex.dur) ~ Ns(tfd,
    knots = kd.rd) + Ns(tfr, knots = kr.rd)
Model 3: cbind(trt(Lx$lex.Cst, Lx$lex.Xst) %in% trnam, Lx$lex.dur) ~ Ns(tfd,
    knots = kd.rd) + Ns(tfr, knots = kr.rd) + Ns(tfd - tfr, knots = ka.rd)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1     50324      10337
2     50321      10260  3   77.541  < 2e-16
3     50319      10253  2    6.898  0.03177
```

We see there is a marginally significant (non-linear) effect of time `at` relapse.

Unfortunately there is currently no `update` method available for `glm.Lexis` (or `gam.Lexis` and `coxph.Lexis`), so we keep the notation with capital letters for the models fitted with `glm.Lexis`, and make a proper `glm` update to interaction models:

```
>  x.rd <- update( m.rd, . ~ . + Ns(      tfr, knots=kr.rd) )
> xx.rd <- update( x.rd, . ~ . + Ns( tfd-tfr, knots=ka.rd) )
> anova( m.rd, x.rd, xx.rd, test="Chisq" )
Analysis of Deviance Table

Model 1: cbind(lex.Xst == "Dead(Rel)", lex.dur) ~ Ns(tfd, knots = kd.rd)
Model 2: cbind(lex.Xst == "Dead(Rel)", lex.dur) ~ Ns(tfd, knots = kd.rd) +
    Ns(tfr, knots = kr.rd)
Model 3: cbind(lex.Xst == "Dead(Rel)", lex.dur) ~ Ns(tfd, knots = kd.rd) +
    Ns(tfr, knots = kr.rd) + Ns(tfd - tfr, knots = ka.rd)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1     50324      10337
2     50321      10260  3   77.541  < 2e-16
3     50319      10253  2    6.898  0.03177
```

What we are doing here is adding interactions between timescales, popularly known as "testing for non-proportionality". Adding time since relapse as a time scale is one extension of the model with proportional mortality rates between persons with and without relapse, by letting the HR depend on time since relapse. A further extension is to add an effect of the difference of the two is yet another interaction term.

The tests are however not particularly relevant; a considerably large dataset as the current may yield statistical significance where no clinically relevant significant effects are present. Therefore, testing of proportionality must necessarily be supported by displays of the *shape* of the interactions.

We can show how the addition of time since relapse and time at relapse affects the estimated mortality by showing mortality after relapse as a function of time since diagnosis for different times of relapse — by showing curves starting at the times of relapse.

We briefly look at the survival in relapse; we see that very few deaths occur after 7 years, so we only draw the predictions till 7 years

```
> with( subset(Sbc,lex.Xst=="Dead(Rel)"), quantile(tfr+lex.dur,37:39/40) )
   92.5%      95%    97.5%
5.505817 6.003559 7.282409
```

```
> nd <- data.frame(expand.grid(tfd = c(NA, seq(0, 15, 0.1)),
+                              tad = c(0, 0.5, 1, 2, 3, 5, 8)))
> nd <- subset(transform(nd, tfr = tfd - tad ),
+              (tfr>=0 & tfr<7) | is.na(tfr) )
> head( nd )
  tfd tad tfr
1  NA   0  NA
2 0.0   0 0.0
3 0.1   0 0.1
4 0.2   0 0.2
5 0.3   0 0.3
6 0.4   0 0.4
> matshade(nd$tfd, cbind(ci.pred( x.rd, nd),
+                        ci.pred(xx.rd, nd)) * 100, plot = TRUE,
+          type = "l", lty = c("11", "solid"), lend = "butt",
+          lwd = 3, col = gray(c(5, 0) / 10), alpha = c(0, 0.07), las = 1,
+          log = "y", xlab = "Time since diagnosis (years)",
+          ylim = c(5,100), ylab = "Mortality rate per 100 PY" )
> matshade(tt <- seq(0,15,0.1), ci.pred( m.rd, data.frame(tfd = tt) )*100,
+          lwd = 3, lty = 1, col = clr[3] )
```

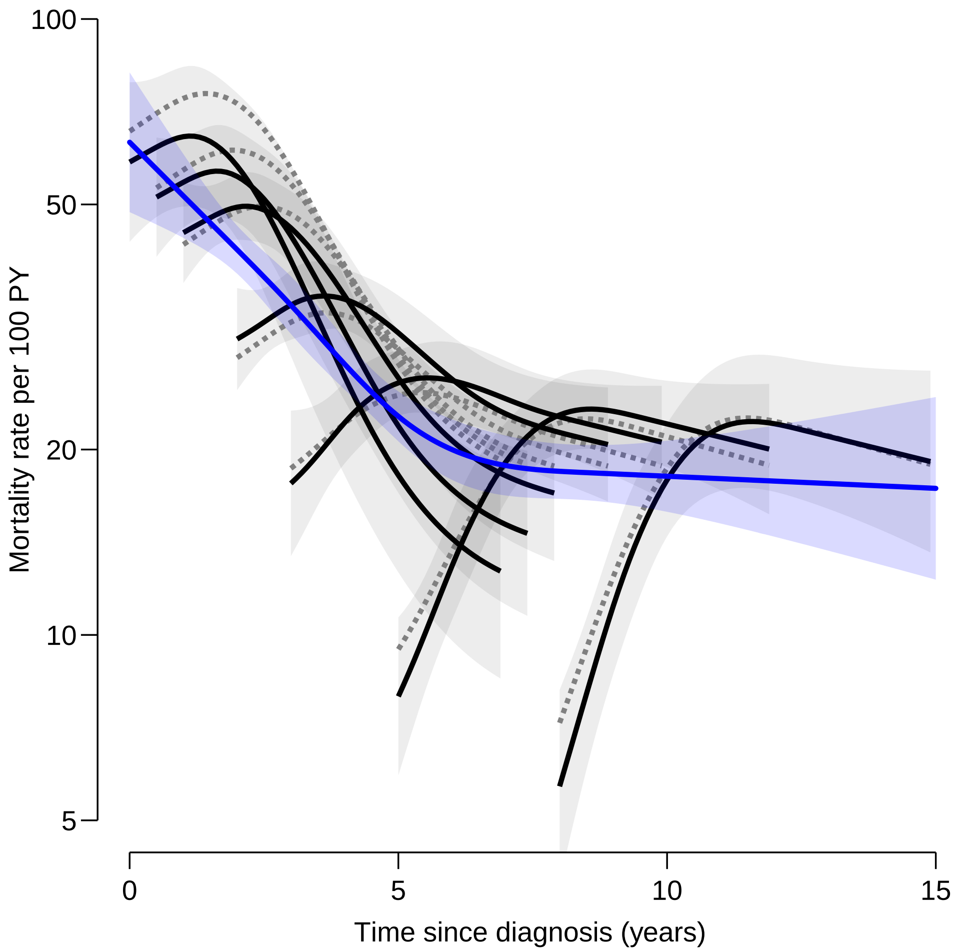

Figure 3: *Estimated mortality among women in relapse. The gray lines represent mortality for women relapsed at 0, 0.5, 1, 2, 3, 5, 8 years after diagnosis. 95% confidence intervals are shown as shades. The full lines are predictions from the model where the time* at *relapse is modeled too (*`xx.rd`*), the broken lines are the rates where only time since diagnosis and time since relapse are included (*`x.rd`*, no c.i. shown). The blue line is from the model where only time since diagnosis is included (*`m.rd`*, "proportional hazards model"), corresponding to the blue line in figure 2.* `./bcMS-rd-int`

From figure 3 we see that the simple model completely misses to describe the initial increase in mortality after relapse, and that the inclusion of time *at* relapse shows that mortality among early relapsees tails off after some 2 years, the faster the earlier the relapse.

4 Including covariates

Following the example in the paper, we include the available covariates in the models:

```
> c.ar  <- update( m.ar, . ~ . + age + size + nodes + pr.tr + hormon)
> c.ad  <- update( m.ad, . ~ . + age + size + nodes + pr.tr + hormon)
> c.rd  <- update( m.rd, . ~ . + age + size + nodes + pr.tr + hormon)
> cx.rd <- update( x.rd, . ~ . + age + size + nodes + pr.tr + hormon)
```

```
> cxx.rd<- update(xx.rd, . ~ . + age + size + nodes + pr.tr + hormon)
```

We can assess to what extent the covariates have been confounding the effects if the timescales by showing the rates for a select set of these covariates in a display similar to the one in figure 3.

```
> nd <- transform(nd,
+                 age = 54,
+                size = "<=20 mm",
+               nodes = 1,
+               pr.tr = 3,
+              hormon = "no")
> head( nd )
  tfd tad tfr age    size nodes pr.tr hormon
1  NA   0  NA  54 <=20 mm     1     3     no
2 0.0   0 0.0  54 <=20 mm     1     3     no
3 0.1   0 0.1  54 <=20 mm     1     3     no
4 0.2   0 0.2  54 <=20 mm     1     3     no
5 0.3   0 0.3  54 <=20 mm     1     3     no
6 0.4   0 0.4  54 <=20 mm     1     3     no
```

With this update we can make exactly the same prediction as for the model without covariates. Note that we defined `age` as the `current` age so that the time from diagnosis will be the effect *in addition* to the aging effect.

```
> matshade(nd$tfd, cbind(ci.pred( cx.rd, nd),
+                       ci.pred(cxx.rd, nd)) * 100, plot = TRUE,
+          type = "l", lty = c("11", "solid"), lend = "butt",
+          lwd = 3, col = gray(c(5, 0) / 10), alpha = c(0, 0.07), las = 1,
+          log = "y", xlab = "Time since diagnosis (years)",
+          ylim = c(5,100), ylab = "Mortality rate per 100 PY" )
> nt <- data.frame(tfd = seq(0, 15, 0.1))
> nt <- transform(nt,
+                 age = 54,
+                size = "<=20 mm",
+               nodes = 1,
+               pr.tr = 3,
+              hormon = "no")
> matshade(nt$tfd, ci.pred(c.rd, nt) * 100,
+          lwd = 3, lty = 1, col = clr[3] )
```

5 Testing for interaction with time

Further, we can now include terms allowing for interaction between covariates and time since diagnosis (often termed "non-proportionality" in the vein of never foregoing an opportunity to invent yet another term for a well-known concept). It is not entirely clear from the models shown in the paper how the non-proportionality is taken into account, but here we have used the product of the variable with log-time + 0.5 years. In total we have 4 models and 5 variables that we can test for interaction with `tfd`, so we set up an array to hold the p-values for the tests.

```
> int.test <- NArray( list( model=c("c.ar","c.ad","c.rd","cx.rd"),
+                            var=c("age","size","nodes","pr.tr","hormon"),
+                           what=c("d.f.","Dev","P") ) )
> str( int.test )
> int.test[1,1,]<-as.numeric(anova( c.ar,update( c.ar,.~.+log(tfd+0.5):age   ),test="Chisq")[2,3:5])
> int.test[1,2,]<-as.numeric(anova( c.ar,update( c.ar,.~.+log(tfd+0.5):size  ),test="Chisq")[2,3:5])
> int.test[1,3,]<-as.numeric(anova( c.ar,update( c.ar,.~.+log(tfd+0.5):nodes ),test="Chisq")[2,3:5])
```

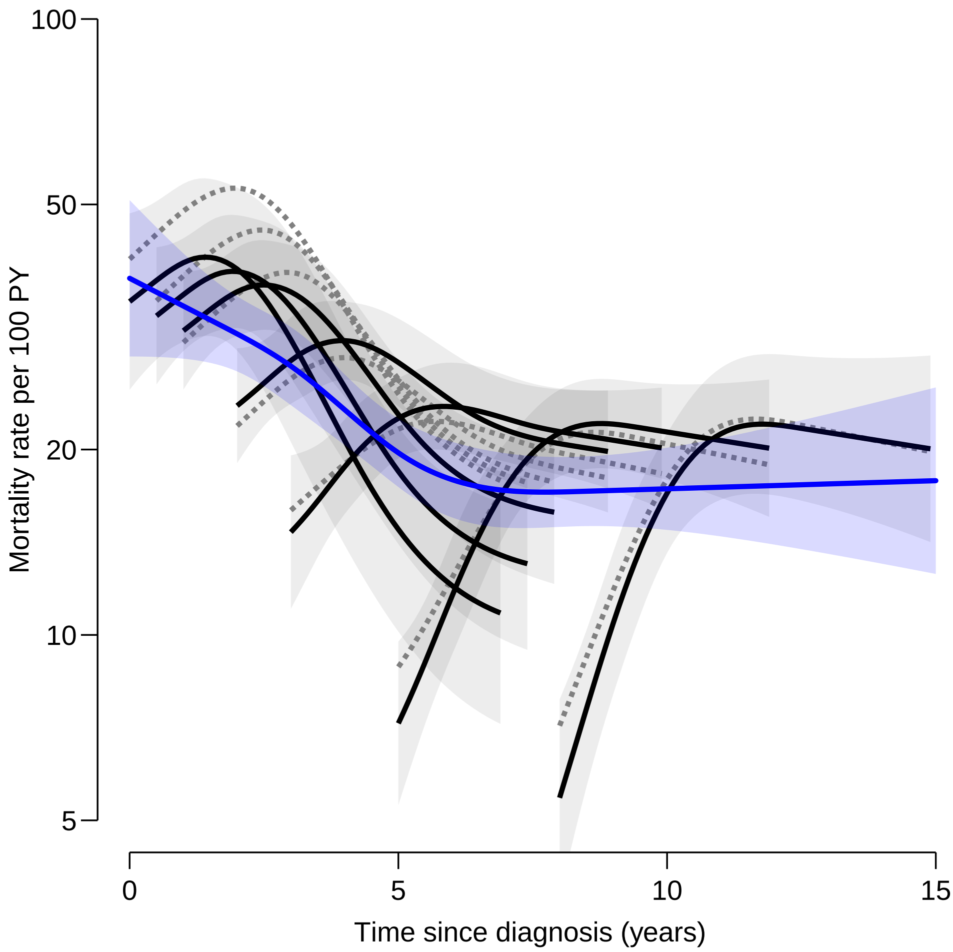

Figure 4: *Estimated mortality among women in relapse from models with covariates* **age**=*54,* **size**=<= 20 *mm,* **nodes**=*1,* **pr.tr**=*3 and* **hormon**=*no. The gray lines represent mortality for women relapsed at 0, 0.5, 1, 2, 3, 5, 8 years after diagnosis. 95% confidence intervals are shown as shades. The full lines are predictions from the model where the time at relapse is modeled too (*`cxx.rd`*), the broken lines are the rates where only time since diagnosis and time since relapse are included (*`cx.rd`*, no c.i. shown). The blue line is from the model where only time since diagnosis is included (*`cm.rd`*, "proportional hazards model"), corresponding to the blue line in figure 2.* `./bcMS-rd-int-cov`

```
> int.test[1,4,]<-as.numeric(anova( c.ar,update( c.ar,.~.+log(tfd+0.5):pr.tr ),test="Chisq")[2,3:5])
> int.test[1,5,]<-as.numeric(anova( c.ar,update( c.ar,.~.+log(tfd+0.5):hormon),test="Chisq")[2,3:5])
> int.test[2,1,]<-as.numeric(anova( c.ad,update( c.ad,.~.+log(tfd+0.5):age   ),test="Chisq")[2,3:5])
> int.test[2,2,]<-as.numeric(anova( c.ad,update( c.ad,.~.+log(tfd+0.5):size  ),test="Chisq")[2,3:5])
> int.test[2,3,]<-as.numeric(anova( c.ad,update( c.ad,.~.+log(tfd+0.5):nodes ),test="Chisq")[2,3:5])
> int.test[2,4,]<-as.numeric(anova( c.ad,update( c.ad,.~.+log(tfd+0.5):pr.tr ),test="Chisq")[2,3:5])
> int.test[2,5,]<-as.numeric(anova( c.ad,update( c.ad,.~.+log(tfd+0.5):hormon),test="Chisq")[2,3:5])
> int.test[3,1,]<-as.numeric(anova( c.rd,update( c.rd,.~.+log(tfd+0.5):age   ),test="Chisq")[2,3:5])
> int.test[3,2,]<-as.numeric(anova( c.rd,update( c.rd,.~.+log(tfd+0.5):size  ),test="Chisq")[2,3:5])
> int.test[3,3,]<-as.numeric(anova( c.rd,update( c.rd,.~.+log(tfd+0.5):nodes ),test="Chisq")[2,3:5])
> int.test[3,4,]<-as.numeric(anova( c.rd,update( c.rd,.~.+log(tfd+0.5):pr.tr ),test="Chisq")[2,3:5])
> int.test[3,5,]<-as.numeric(anova( c.rd,update( c.rd,.~.+log(tfd+0.5):hormon),test="Chisq")[2,3:5])
> int.test[4,1,]<-as.numeric(anova(cx.rd,update(cx.rd,.~.+log(tfd+0.5):age   ),test="Chisq")[2,3:5])
> int.test[4,2,]<-as.numeric(anova(cx.rd,update(cx.rd,.~.+log(tfd+0.5):size  ),test="Chisq")[2,3:5])
> int.test[4,3,]<-as.numeric(anova(cx.rd,update(cx.rd,.~.+log(tfd+0.5):nodes ),test="Chisq")[2,3:5])
```

```
> int.test[4,4,]<-as.numeric(anova(cx.rd,update(cx.rd,.~.+log(tfd+0.5):pr.tr ),test="Chisq")[2,3:5])
> int.test[4,5,]<-as.numeric(anova(cx.rd,update(cx.rd,.~.+log(tfd+0.5):hormon),test="Chisq")[2,3:5])
> save( int.test, file="int-test.Rda")
```

```
> load( file="/home/bendix/teach/AdvCoh/00/examples/bcMS/int-test.Rda")
> round( int.test[,,2], 2 )

       var
model    age  size nodes pr.tr hormon
  c.ar  3.43 81.32  2.60 77.04  55.67
  c.ad  0.78  1.10  3.04  3.66   0.80
  c.rd  2.92  3.04  2.57 23.35   4.99
  cx.rd 3.24  3.28  2.81 21.67     NA

> round( int.test[,,3], 4 )

       var
model      age   size  nodes  pr.tr hormon
  c.ar  0.0639 0.0000 0.1070 0.0000 0.0000
  c.ad  0.3763 0.7760 0.0814 0.0559 0.6710
  c.rd  0.0874 0.3854 0.1086 0.0000 0.0827
  cx.rd 0.0718 0.3506 0.0936 0.0000     NA

> round( int.test[,,1], 0 )

       var
model   age size nodes pr.tr hormon
  c.ar    1    3     1     1      2
  c.ad    1    3     1     1      2
  c.rd    1    3     1     1      2
  cx.rd   1    3     1     1     NA
```

Thus it seems that there are interactions between time from diagnosis and progesterone for all transition rates, and that relapse rates additionally have interactions between time from diagnosis and size and hormone therapy. The p-values would of course have looked slightly differently if some other parametric shape of the interactions were chosen. This is merely a reflection of the fact that there is no well-defined concept of test for proportionality; as in all cases of interaction with at least one quantitative variable involved the test for interaction is always a test versus some pre-specified alternative in the form of a specific *shape* of the interaction.

5.1 The interaction models (non-proportionality)

It is bad practice to make interaction tests without showing how the interactions look; however this is not a trivial task with three different interactions, but if you do not bother to show the shape and size of estimated interactions, then it is presumably better to refrain from interaction tests in the first place.

So we include the identified interactions in the models for the rates. Note that we also for the sake of notational convenience also include a void update of the model for mortality after relapse where we take time since relapse into account:

```
>  i.ar <- update( c.ar, . ~ . + log(tfd+0.5):size
+                              + log(tfd+0.5):pr.tr
+                              + log(tfd+0.5):hormon)
>  i.ad <- c.ad
>  i.rd <- update( c.rd, . ~ . + log(tfd+0.5):pr.tr)
> ix.rd <- update(cx.rd, . ~ . + log(tfd+0.5):pr.tr)
> round(ci.lin(i.ad ), 4)
```

```
                           Estimate StdErr        z      P     2.5%    97.5%
(Intercept)                -13.5764 0.6005 -22.6084 0.0000 -14.7534 -12.3995
Ns(tfd, knots = kd.ad)1      0.2873 0.2608   1.1019 0.2705  -0.2237   0.7984
Ns(tfd, knots = kd.ad)2      1.9852 0.2804   7.0809 0.0000   1.4357   2.5347
Ns(tfd, knots = kd.ad)3      1.1706 0.1944   6.0215 0.0000   0.7896   1.5516
age                          0.1286 0.0081  15.8754 0.0000   0.1128   0.1445
size>20-50 mm                0.1714 0.1610   1.0644 0.2871  -0.1442   0.4869
size>50 mm                   0.4069 0.2330   1.7465 0.0807  -0.0497   0.8636
nodes                        0.0444 0.0184   2.4150 0.0157   0.0084   0.0804
pr.tr                        0.0305 0.0336   0.9069 0.3645  -0.0354   0.0963
hormonyes                   -0.0955 0.2312  -0.4131 0.6795  -0.5486   0.3576
> round(ci.lin(i.ar ), 4)
                                Estimate StdErr        z      P     2.5%   97.5%
(Intercept)                      -2.9449 0.1964 -14.9979 0.0000  -3.3297 -2.5600
Ns(tfd, knots = kd.ar)1          -4.6099 0.5477  -8.4167 0.0000  -5.6834 -3.5364
Ns(tfd, knots = kd.ar)2          -8.0623 1.1289  -7.1419 0.0000 -10.2748 -5.8498
Ns(tfd, knots = kd.ar)3          -5.7271 0.6743  -8.4932 0.0000  -7.0487 -4.4055
age                              -0.0061 0.0021  -2.9224 0.0035  -0.0103 -0.0020
size>20-50 mm                     0.7402 0.1153   6.4223 0.0000   0.5143  0.9661
size>50 mm                        1.1455 0.1503   7.6200 0.0000   0.8508  1.4401
nodes                             0.0783 0.0045  17.2651 0.0000   0.0695  0.0872
pr.tr                            -0.1880 0.0218  -8.6069 0.0000  -0.2309 -0.1452
hormonyes                        -0.3157 0.1497  -2.1089 0.0350  -0.6092 -0.0223
size<=20 mm:log(tfd + 0.5)        3.4405 0.5083   6.7685 0.0000   2.4442  4.4368
size>20-50 mm:log(tfd + 0.5)      3.1347 0.5043   6.2154 0.0000   2.1462  4.1232
size>50 mm:log(tfd + 0.5)         2.9695 0.5082   5.8432 0.0000   1.9735  3.9656
pr.tr:log(tfd + 0.5)              0.1305 0.0170   7.6747 0.0000   0.0972  0.1639
hormonyes:log(tfd + 0.5)          0.2472 0.1224   2.0195 0.0434   0.0073  0.4871
> round(ci.lin(i.rd ), 4)
                           Estimate StdErr       z      P    2.5%   97.5%
(Intercept)                 -0.9357 0.1568 -5.9666 0.0000 -1.2431 -0.6284
Ns(tfd, knots = kd.rd)1     -0.8855 0.1251 -7.0781 0.0000 -1.1306 -0.6403
Ns(tfd, knots = kd.rd)2     -1.3036 0.1670 -7.8076 0.0000 -1.6309 -0.9764
Ns(tfd, knots = kd.rd)3     -0.9527 0.1242 -7.6710 0.0000 -1.1961 -0.7093
age                          0.0049 0.0024  2.0239 0.0430  0.0002  0.0096
size>20-50 mm                0.1654 0.0712  2.3217 0.0202  0.0258  0.3050
size>50 mm                   0.3266 0.0993  3.2891 0.0010  0.1320  0.5212
nodes                        0.0296 0.0058  5.1389 0.0000  0.0183  0.0409
pr.tr                       -0.2771 0.0396 -7.0011 0.0000 -0.3547 -0.1996
hormonyes                    0.0432 0.0975  0.4429 0.6578 -0.1478  0.2342
pr.tr:log(tfd + 0.5)         0.1156 0.0245  4.7207 0.0000  0.0676  0.1635
> round(ci.lin(ix.rd), 4)
                           Estimate StdErr        z      P    2.5%   97.5%
(Intercept)                 -1.0005 0.1583  -6.3198 0.0000 -1.3108 -0.6902
Ns(tfd, knots = kd.rd)1     -1.3815 0.1435  -9.6239 0.0000 -1.6629 -1.1002
Ns(tfd, knots = kd.rd)2     -2.1705 0.2035 -10.6634 0.0000 -2.5695 -1.7716
Ns(tfd, knots = kd.rd)3     -1.3495 0.1431  -9.4320 0.0000 -1.6300 -1.0691
Ns(tfr, knots = kr.rd)1      0.8184 0.1258   6.5077 0.0000  0.5719  1.0649
Ns(tfr, knots = kr.rd)2      1.2898 0.1746   7.3858 0.0000  0.9475  1.6321
Ns(tfr, knots = kr.rd)3      0.6399 0.1112   5.7538 0.0000  0.4219  0.8578
age                          0.0043 0.0024   1.7698 0.0768 -0.0005  0.0090
size>20-50 mm                0.1395 0.0715   1.9522 0.0509 -0.0006  0.2796
size>50 mm                   0.2880 0.0996   2.8927 0.0038  0.0929  0.4831
nodes                        0.0274 0.0058   4.7106 0.0000  0.0160  0.0388
pr.tr                       -0.2716 0.0395  -6.8838 0.0000 -0.3489 -0.1943
hormonyes                    0.0882 0.0982   0.8984 0.3689 -0.1042  0.2807
pr.tr:log(tfd + 0.5)         0.1112 0.0244   4.5625 0.0000  0.0634  0.1590
```

Note that we have one aliased parameter (`NA` for `z` and `P`) in the model with effects of the two timescales (`tfd`, `tfr`) and their difference. This is because the natural spline parametrization include the linear effects of the variables modeled. This has no effect of the predictions however; and these are the only ones we are concerned about.

In the following we shall use reference values for each of the covariates, and show mortality rates as function of time since diagnosis for select values of the interaction variables:

For each of the three covariates with interactions we construct a prediction frame with varying levels of the interaction variables:

```
> nd.size <- data.frame(tfd = rep( c(NA,seq(0,15,0.1)), 3 ),
+                  age = 45,
+                 size = rep( levels(Lbc$size), each=152 ),
+                nodes = 5,
+                pr.tr = 3,
+               hormon = levels(Lbc$hormon)[1] )
> nd.pr <- data.frame(tfd = rep( c(NA,seq(0,15,0.1)), 6 ),
+                  age = 45,
+                 size = levels(Lbc$size)[2],
+                nodes = 5,
+                pr.tr = rep( 0:5, each=152 ),
+               hormon = levels(Lbc$hormon)[1] )
> nd.hormon <- data.frame(tfd = rep( c(NA,seq(0,15,0.1)), 2 ),
+                  age = 45,
+                 size = levels(Lbc$size)[2],
+                nodes = 5,
+                pr.tr = 3,
+               hormon = rep( levels(Lbc$hormon), each=152 ) )
```

For each of these prediction frames we can plot the three estimated transition rates as we did for the overall rates (or rather the rates estimated using only the `tfd` variable as covariate). Moreover we will plot the estimated rates both from the interaction models (`i.`) and the main-effects models (`c.`):

```
> clr <- rainbow(3) ; yl <- c(0.03,60)
> ad.c.rate <- ci.pred(c.ad, nd.size)
> ad.i.rate <- ci.pred(i.ad, nd.size)
> ar.c.rate <- ci.pred(c.ar, nd.size)
> ar.i.rate <- ci.pred(i.ar, nd.size)
> rd.c.rate <- ci.pred(c.rd, nd.size)
> rd.i.rate <- ci.pred(i.rd, nd.size)
> matplot(nd.size$tfd, cbind(ad.c.rate, ad.i.rate,
+                            ar.c.rate, ar.i.rate,
+                            rd.c.rate, rd.i.rate) * 100,
+         type = "l", lty = rep(c("22", "solid"),each = 3),
+         lwd = c(2,0,0), col = rep(clr, each = 6), las = 1, lend = "butt",
+         log = "y", xlab = "Time since diagnosis (years)",
+         ylim = yl, ylab = "Rate per 100 PY" )
> text( par("usr")[2]*0.95, (10^par("usr"))[3]*1.4^(1:3),
+       c("A->D","A->R","R->D"), col = clr, adj = 1, font = 2 )
```

```
> ad.c.rate <- ci.pred(c.ad, nd.pr)
> ad.i.rate <- ci.pred(i.ad, nd.pr)
> ar.c.rate <- ci.pred(c.ar, nd.pr)
> ar.i.rate <- ci.pred(i.ar, nd.pr)
> rd.c.rate <- ci.pred(c.rd, nd.pr)
> rd.i.rate <- ci.pred(i.rd, nd.pr)
> matplot(nd.pr$tfd, cbind(ad.c.rate, ad.i.rate,
+                          ar.c.rate, ar.i.rate,
+                          rd.c.rate, rd.i.rate) * 100,
+         type = "l", lty = rep(c("22", "solid"),each = 3),
+         lwd = c(2,0,0), col = rep(clr, each = 6), las = 1, lend = "butt",
+         log = "y", xlab = "Time since diagnosis (years)",
+         ylim = yl, ylab = "Rate per 100 PY" )
> text( par("usr")[2]*0.95, (10^par("usr"))[3]*1.4^(1:3),
+       c("A->D","A->R","R->D"), col = clr, adj = 1, font = 2 )
```

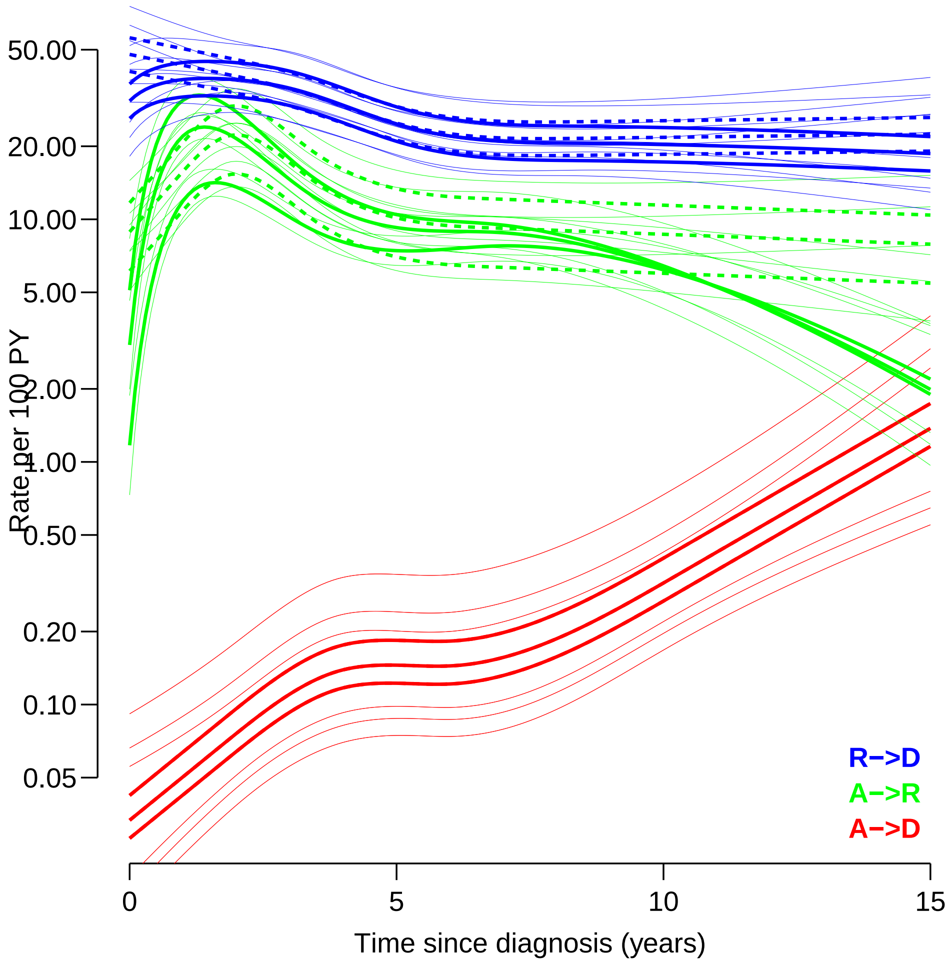

Figure 5: *Transition rates as function of time since diagnosis; the broken lines are from the main effects models and the full lines from the interaction model with* `age`*=54,* `nodes`*=5,* `pr.tr`*=3,* `hormon`*=no and where* `size` *assumes the values < 20 mm, 20–50 mm and > 50 mm (only the Alive→Rel transition). Thus the test of interaction is the comparison of the sets of parallel broken lines with the non-parallel full lines.* `./bcMS-int-size`

```
> ad.c.rate <- ci.pred(c.ad, nd.hormon)
> ad.i.rate <- ci.pred(i.ad, nd.hormon)
> ar.c.rate <- ci.pred(c.ar, nd.hormon)
> ar.i.rate <- ci.pred(i.ar, nd.hormon)
> rd.c.rate <- ci.pred(c.rd, nd.hormon)
> rd.i.rate <- ci.pred(i.rd, nd.hormon)
> matplot(nd.hormon$tfd, cbind(ad.c.rate, ad.i.rate,
+                               ar.c.rate, ar.i.rate,
+                               rd.c.rate, rd.i.rate) * 100,
+         type = "l", lty = rep(c("22","solid"),each = 3),
+         lwd = c(2,0,0), col = rep(clr, each = 6), las = 1,
+         lend = "butt", log = "y", ylim = yl, ylab = "Rate per 100 PY",
+         xlab = "Time since diagnosis (years)")
> text(par("usr")[2]*0.95, (10^par("usr"))[3]*1.4^(1:3),
+      c("A->D","A->R","R->D"), col = clr, adj = 1, font = 2 )
```

The general picture from the figures 5, 6 and 7 is that the major interactions are with the relapse rates, where it seems that the interactions mainly reveal that the major effects are early, and are possibly even reversed later.

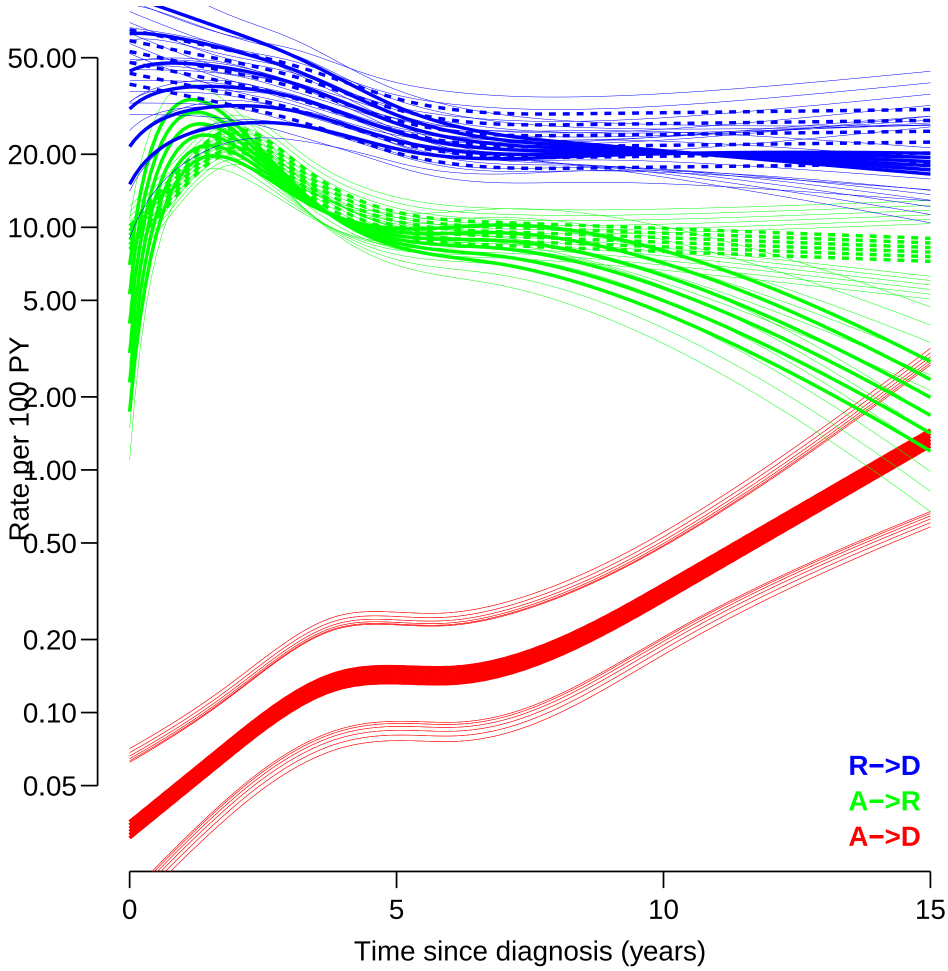

Figure 6: *Transition rates as function of time since diagnosis, the broken lines are from the main effects models and the full lines from the interaction model with* `age`*=54,* `size`*=20–50 mm,* `nodes`*=5,* `hormon`*=no and where* `pr.tr` *assumes the values 0–6. Thus the test of interaction is the comparison of the sets of parallel broken lines with the non-parallel full lines — no interaction for the Alive→Dead transition.* `./bcMS-int-pr`

This is merely to illustrate how the usual largely uninformative "test of proportionality" necessarily must be complemented by graphical displays of the non-proportionalities so that it in `substantial` terms can be assessed whether the interactions are of relevance or not.

6 Predicting state occupancy

As done in the SiM paper [1] we predict state occupancy for a patient aged 54, with a transformed progesterone level of 3, and no hormone therapy (?), for different tumour groups and node numbers 0, 10 and 20. We shall also compute the expected time alive, so the calculations will be made for node numbers 0, 1, 5 and 10 — a reasonable set of values seen in the dataset, with 0 being the by far most prevalent number.

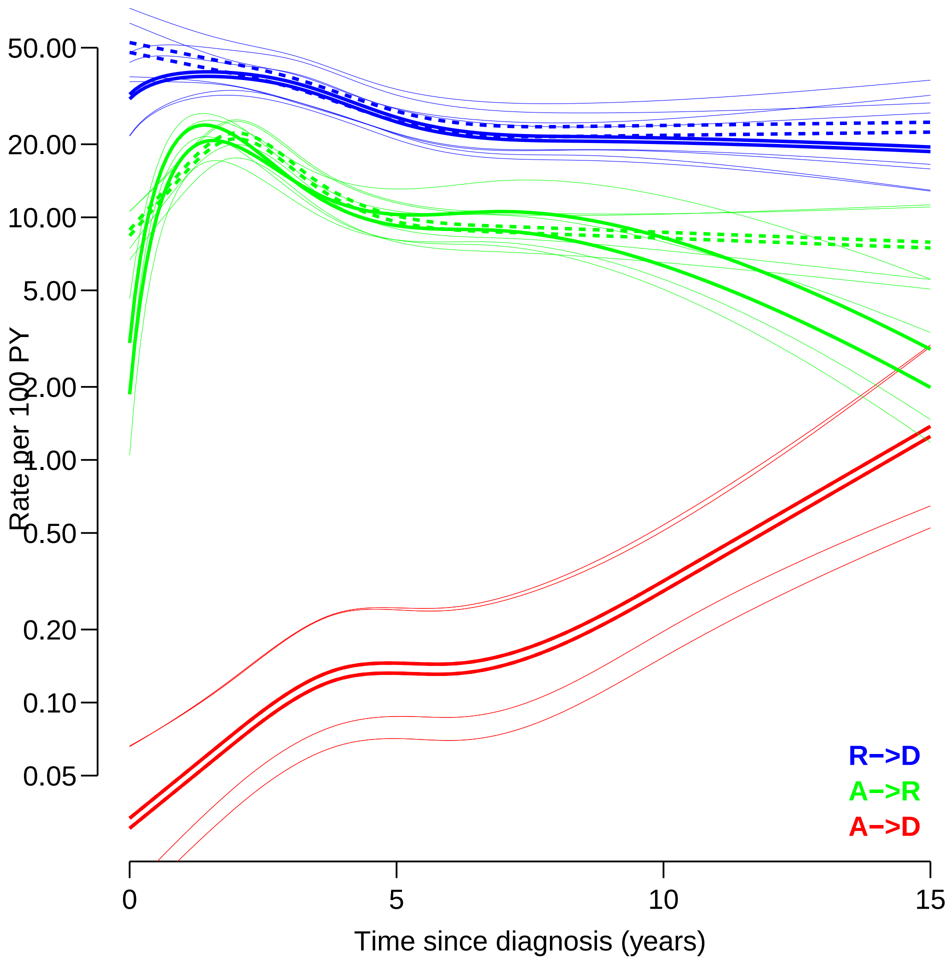

Figure 7: *Transition rates as function of time since diagnosis, the broken lines are from the main effects models and the full lines from the interaction model with* `age`*=54,* `size`*=20–50 mm,* `nodes`*=5,* `pr.tr`*=3 and where* `hormon` *assumes the values no and yes. Thus the test of interaction is the comparison of the sets of parallel broken lines with the non-parallel full lines.* `./bcMS-int-hormon`

6.1 Initial cohort

To this end we construct a Lexis object from `Rbc`; the main thing here is to maintain the `Lexis`-specific attributes which will be used in the simulation process. And all the time scale variables too, even if `A` and `P` will not be used in the simulation (because they are not in any of the models) — the latter is a feature (or bug) in `simLexis`; the function will refer to all timescales in the object even if they are not in the models and hence not explicitly used in the calculations:

```
> names( Rbc )
 [1] "tfd"      "A"        "P"        "tfr"      "lex.dur" "lex.Cst" "lex.Xst" "lex.id"
 [9] "pid"      "year"     "age"      "meno"     "size"     "grade"   "nodes"   "pr"
[17] "pr.tr"    "er"       "hormon"   "chemo"    "tor"      "tom"     "tod"     "tox"
[25] "xst"
```

```
> Lini <- Rbc[NULL,c("tfd","A","P","tfr",
+                    "lex.Cst","lex.Xst","lex.dur","lex.id",
+                    "age","size","nodes","pr.tr","hormon")]
```

```
> pr.nodes <- c(0,1,5,10)
> npr <- nlevels(Rbc$size) * length(pr.nodes)
> Lini[1:npr,"tfd"] <- 0
> Lini[1:npr,"tfr"] <- NA
> Lini[1:npr,"lex.Cst"] <- "Alive"
> Lini[1:npr,"age"] <- 54
> Lini[1:npr,"size"] <- rep(levels(Rbc$size), length(pr.nodes))
> Lini[1:npr,"nodes"] <- rep(pr.nodes, each=nlevels(Rbc$size))
> Lini[1:npr,"pr.tr"] <- 3
> Lini[1:npr,"hormon"] <- "no"
> Lini
```

```
   tfd  A  P tfr lex.Cst lex.Xst lex.dur lex.id age      size nodes pr.tr hormon
1    0 NA NA  NA   Alive    <NA>      NA     NA  54    <=20 mm     0     3     no
2    0 NA NA  NA   Alive    <NA>      NA     NA  54 >20-50 mm     0     3     no
3    0 NA NA  NA   Alive    <NA>      NA     NA  54     >50 mm     0     3     no
4    0 NA NA  NA   Alive    <NA>      NA     NA  54    <=20 mm     1     3     no
5    0 NA NA  NA   Alive    <NA>      NA     NA  54 >20-50 mm     1     3     no
6    0 NA NA  NA   Alive    <NA>      NA     NA  54     >50 mm     1     3     no
7    0 NA NA  NA   Alive    <NA>      NA     NA  54    <=20 mm     5     3     no
8    0 NA NA  NA   Alive    <NA>      NA     NA  54 >20-50 mm     5     3     no
9    0 NA NA  NA   Alive    <NA>      NA     NA  54     >50 mm     5     3     no
10   0 NA NA  NA   Alive    <NA>      NA     NA  54    <=20 mm    10     3     no
11   0 NA NA  NA   Alive    <NA>      NA     NA  54 >20-50 mm    10     3     no
12   0 NA NA  NA   Alive    <NA>      NA     NA  54     >50 mm    10     3     no
```

```
> str( Lini )
```

```
Classes ‘Lexis’ and 'data.frame':        12 obs. of  13 variables:
 $ tfd    : num  0 0 0 0 0 0 0 0 0 0 ...
 $ A      : num  NA NA NA NA NA NA NA NA NA NA ...
 $ P      : num  NA NA NA NA NA NA NA NA NA NA ...
 $ tfr    : num  NA NA NA NA NA NA NA NA NA NA ...
 $ lex.Cst: Factor w/ 4 levels "Alive","Rel",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ lex.Xst: Factor w/ 4 levels "Alive","Rel",..: NA NA NA NA NA NA NA NA NA NA ...
 $ lex.dur: num  NA NA NA NA NA NA NA NA NA NA ...
 $ lex.id : int  NA NA NA NA NA NA NA NA NA NA ...
 $ age    : num  54 54 54 54 54 54 54 54 54 54 ...
 $ size   : Factor w/ 3 levels "<=20 mm",">20-50 mm",..: 1 2 3 1 2 3 1 2 3 1 ...
 $ nodes  : num  0 0 0 1 1 1 5 5 5 10 ...
 $ pr.tr  : num  3 3 3 3 3 3 3 3 3 3 ...
 $ hormon : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
 - attr(*, "time.scales")= chr  "tfd" "A" "P" "tfr"
 - attr(*, "time.since")= chr  "" "" "" "Rel"
 - attr(*, "breaks")=List of 4
  ..$ tfd: NULL
  ..$ A  : NULL
  ..$ P  : NULL
  ..$ tfr: NULL
```

6.2 Transition rates

In order to simulate a number of persons initiating follow-up (=diagnosed with breast cancer) with these covariate patterns according to our model, we must also define the transition objects (that is, specify models for the three transition rates) — we make one designed to mimic the models used in the SiM paper [1] and one using the better fitting model for death after relapse—the difference is the model used for the mortality after relapse:

```
> TR  <- list( Alive = list( Dead = i.ad,
+                             Rel = i.ar ),
+                Rel = list( "Dead(Rel)" = i.rd ) )
> TRx <- list( Alive = list( Dead = i.ad,
```

```
+                              Rel = i.ar ),
+                 Rel = list( "Dead(Rel)" = ix.rd ) )
> lapply( TR, names )
$Alive
[1] "Dead" "Rel"

$Rel
[1] "Dead(Rel)"

> lapply( TR, lapply, class )
$Alive
$Alive$Dead
[1] "glm" "lm"

$Alive$Rel
[1] "glm" "lm"


$Rel
$Rel$`Dead(Rel)`
[1] "glm" "lm"
```

6.3 Simulation of a large cohort

With this in place we can simulate:

```
> system.time( xx <- simLexis( Tr=TR , init=Lini, N=200, t.range=16 ) )
   user  system elapsed
  5.123   0.685   4.973
```

(There were some problems causing a crash when trying to simulate 10,000 persons in one go, so we did things in chunks).

```
> # not evaluated, run interactively before final compilation
> set.seed( 1952 )
> x0 <- simLexis(Tr = TR , init = Lini, N = 2000, t.range = 16)
> x1 <- simLexis(Tr = TR , init = Lini, N = 2000, t.range = 16)
> x2 <- simLexis(Tr = TR , init = Lini, N = 2000, t.range = 16)
> x3 <- simLexis(Tr = TR , init = Lini, N = 2000, t.range = 16)
> x4 <- simLexis(Tr = TR , init = Lini, N = 2000, t.range = 16)
> sL <- rbind(x0, transform(x1, lex.id = lex.id+25000 ),
+                 transform(x2, lex.id = lex.id+50000 ),
+                 transform(x3, lex.id = lex.id+75000 ),
+                 transform(x4, lex.id = lex.id+100000))
> summary( sL )
> s0 <- simLexis(Tr = TRx, init = Lini, N = 2000, t.range = 16)
> s1 <- simLexis(Tr = TRx, init = Lini, N = 2000, t.range = 16)
> s2 <- simLexis(Tr = TRx, init = Lini, N = 2000, t.range = 16)
> s3 <- simLexis(Tr = TRx, init = Lini, N = 2000, t.range = 16)
> s4 <- simLexis(Tr = TRx, init = Lini, N = 2000, t.range = 16)
> sLx<- rbind(s0, transform(s1, lex.id = lex.id+25000 ),
+                 transform(s2, lex.id = lex.id+50000 ),
+                 transform(s3, lex.id = lex.id+75000 ),
+                 transform(s4, lex.id = lex.id+100000))
> summary( sLx )
> save(sL, sLx,
+      file = "/home/bendix/teach/AdvCoh/00/examples/bcMS/sL.Rda")
```

We asked for simulation of 10,000 persons with each of the 12 covariate patterns in `Lini`, a total of 120,000 persons:

```
> load(file = "/home/bendix/teach/AdvCoh/00/examples/bcMS/sL.Rda")
> summary(sLx)
Transitions:
     To
From    Alive   Rel Dead Dead(Rel)  Records:  Events: Risk time:  Persons:
  Alive 30830 81458 7712         0    120000    89170   902265.3    120000
  Rel       0  9428    0     72030     81458    72030   280698.4     81458
  Sum   30830 90886 7712     72030    201458   161200  1182963.7    120000
```

6.4 State occupancy probabilities

We can now devise the state probabilities by using `nState` and `pState` — here we just use an arbitrary subset to get the object structure:

```
> nn <- nState( sLx[1:1000,], at=seq(0,16,0.1), from=0, time.scale="tfd" )
> pp <- pState( nn, perm=c(1,2,4,3) )
> str( pp )
 'pState' num [1:161, 1:4] 1 1 0.996 0.991 0.987 ...
 - attr(*, "dimnames")=List of 2
  ..$ when : chr [1:161] "0" "0.1" "0.2" "0.3" ...
  ..$ State: chr [1:4] "Alive" "Rel" "Dead(Rel)" "Dead"
```

However this is not what we want; we want the calculation for the 12 different combinations of node and size; so we devise these levels too:

```
> (tt <- with(sLx, table(nodes, size)))
     size
nodes <=20 mm >20-50 mm >50 mm
   0     14923     15773  16443
   1     15266     16122  16728
   5     16270     17259  17777
   10    17607     18457  18833
> prX <- prA <- NArray(c(dimnames(tt), dimnames(pp)))
> str(prA)
 logi [1:4, 1:3, 1:161, 1:4] NA NA NA NA NA NA ...
 - attr(*, "dimnames")=List of 4
  ..$ nodes: chr [1:4] "0" "1" "5" "10"
  ..$ size : chr [1:3] "<=20 mm" ">20-50 mm" ">50 mm"
  ..$ when : chr [1:161] "0" "0.1" "0.2" "0.3" ...
  ..$ State: chr [1:4] "Alive" "Rel" "Dead(Rel)" "Dead"
```

So now we have two arrays to hold the state occupancy probabilities for all combinations of nodes, size and time from diagnosis; thus we need a loop over the 15 subsets to devise the relevant probabilities and put them in the arrays:

```
> for( nn in dimnames(prA)[[1]] )
+ for( ss in dimnames(prA)[[2]] )
+    {
+ prA[nn,ss,,] <- pState( nState( subset( sL , nodes==as.numeric(nn) &
+                                                size==ss ),
+                                  at = seq(0,16,0.1),
+                                from = 0,
+                          time.scale = "tfd" ),
+                          perm = c(1,2,4,3) )
+ prX[nn,ss,,] <- pState( nState( subset( sLx, nodes==as.numeric(nn) &
+                                                size==ss ),
+                                  at = seq(0,16,0.1),
+                                from = 0,
+                          time.scale = "tfd" ),
+                          perm = c(1,2,4,3) )
+    }
> save(prA, prX, file = "pr.Rda")
```

With this array of probabilities we can now plot the state occupancy probabilities as a function of time:

```
> load(file = "/home/bendix/teach/AdvCoh/00/examples/bcMS/pr.Rda")
> clr <- c("forestgreen","maroon")
> clr <- cbind(clr, adjustcolor(clr[2:1], 0.5))
> par(mfcol = c(3,4), mar = c(1,1.5,1,1),
+     mgp = c(3,1,0)/1.6, oma = c(2,2,2,2), las = 1, bty="n")
> nnn <- dimnames(prA)[[1]]
> sss <- dimnames(prA)[[2]]
> for( nn in nnn )
+ for( ss in sss )
+    {
+    plot.pState(prX[nn,ss,,],
+                col = clr, xlim = c(0,15), ylab = "", xlab = "" )
+    lines(as.numeric(dimnames(prX)[[3]]), prX[nn,ss,,  2],
+          lwd = 3, lty = 1, col = "black" )
+ matlines(as.numeric(dimnames(prA)[[3]]), prA[nn,ss,,1:3],
+          lwd = 1, lty = 1, col = "white" )
+    axis(side = 2, at = 0:10/10, labels = NA, tcl = -0.4)
+    axis(side = 4, at = 0:10/10, labels = NA, tcl = -0.4)
+    axis(side = 2, at = 0:50/50, labels = NA, tcl = -0.2)
+    axis(side = 4, at = 0:50/50, labels = NA, tcl = -0.2)
+    }
> mtext(paste( "Nodes =",nnn),
+       side = 3, at = (1:4*2-1)/8, outer = TRUE,
+       line = 0, cex = 0.66, las = 0 )
> mtext(paste( "Size"  ,sss),
+       side = 4, at = (3:1*2-1)/6, outer = TRUE,
+       line = 0, cex = 0.66, las = 0 )
> mtext("Time since diagnosis (years)",
+       side = 1, outer = TRUE, line = 1, cex = 0.66, las = 0 )
> mtext("Probability",
+       side = 2, outer = TRUE, line = 1, cex = 0.66, las = 0 )
```

From figure 8 we see that the interaction model does not change the cumulative measures a lot, and also that the coloured areas have pretty much the same size, with or without the interactions. Also note that the only difference is for the overall survival; the models only differ in the mortality of the relapsed patients, so the probability of being alive without relapse and the probability of being dead without relapse is modeled exactly the same way in the two models.

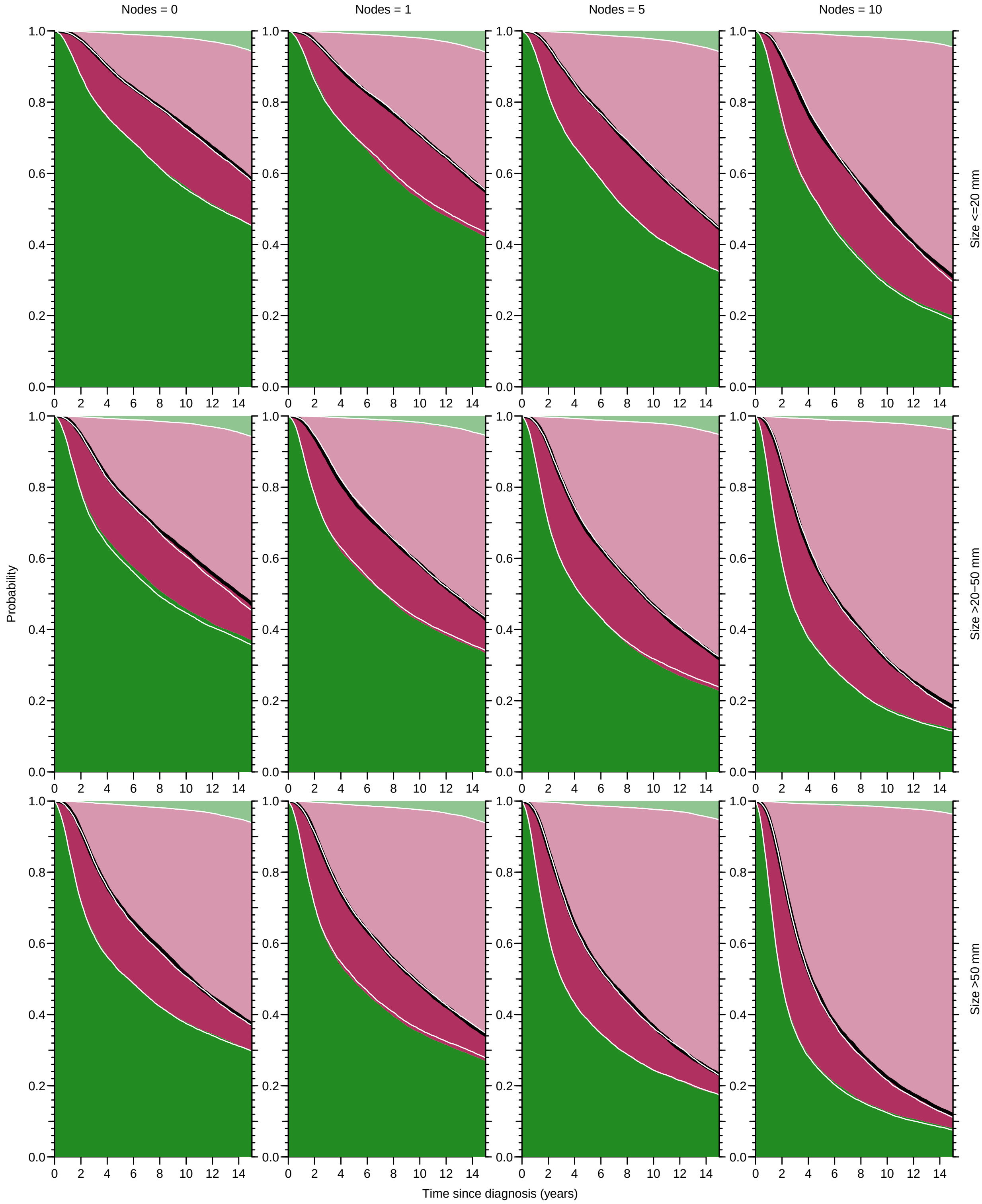

Figure 8: *Probabilities of being alive without relapse (green), with relapse (purple), dead after relapse (light purple), and dead without relapse (light green) The black line is the estimated survival curve. Computed from the model with effects of time since diagnosis as well as since relapse. The white lines indicates what would have been obtained with the model with only time since diagnosis, that is plots corresponding to those in the SiM paper [1].*`../bcMS-states`

7 Years lived with and without relapse

We have the estimated probabilities from the simulation in the arrays `prA`, respectively `prX`. If we want to compute the years lived during the first 15 years, we want the integral under the curves. To this end we need a function that does the triangulation of the area. Here we compute the area under the curves up til 15 years past diagnosis; first based on the naive models, then on the models taking time since relapse into account:

```
> intgr <- function(y, x) (sum(y[-1]) + sum(y[-length(y)])) / 2 * x
> cA <- apply(prA[,,1:151,1:3], c(1,2,4), intgr, 0.1)
> cA[,,3] <- cA[,,2] - cA[,,1]
> dimnames( cA )[[3]] <- c("noRel","Total","Rel")
> cA <- cA[,,c(1,3,2)]
> round(ftable(cA, row.vars = c(3,2)), 2)
```

```
                nodes     0     1     5    10
State size
noRel <=20 mm          9.94  9.71  8.45  6.67
      >20-50 mm        8.41  8.22  6.74  4.82
      >50 mm           7.42  7.19  5.65  3.84
Rel   <=20 mm          2.05  2.10  2.36  2.61
      >20-50 mm        2.23  2.21  2.37  2.50
      >50 mm           2.08  2.08  2.15  2.13
Total <=20 mm         11.99 11.81 10.81  9.28
      >20-50 mm       10.63 10.44  9.12  7.32
      >50 mm           9.50  9.27  7.80  5.97
```

```
> cX <- apply( prX[,,1:151,1:3], c(1,2,4), intgr, 0.1)
> cX[,,3] <- cX[,,2] - cX[,,1]
> dimnames( cX )[[3]] <- c("noRel","Total","Rel")
> cX <- cX[,,c(1,3,2)]
> round(ftable(cX, row.vars = c(3,2)), 2)
```

```
                nodes     0     1     5    10
State size
noRel <=20 mm          9.96  9.61  8.45  6.71
      >20-50 mm        8.58  8.14  6.67  4.88
      >50 mm           7.45  7.06  5.62  3.91
Rel   <=20 mm          2.07  2.17  2.34  2.58
      >20-50 mm        2.18  2.22  2.41  2.44
      >50 mm           2.12  2.18  2.21  2.15
Total <=20 mm         12.03 11.78 10.79  9.29
      >20-50 mm       10.76 10.36  9.08  7.32
      >50 mm           9.57  9.24  7.83  6.06
```

Thus it is clear that both the number of nodes and the tumour size influences the expected lifetime during the first 15 years, although they primarily influence the relapse-free years lived; the years lived with relapse is not that much affected.

Furthermore, we can show the differences between the two sets of models used, both in years and percentwise difference:

```
> round(ftable( cX - cA          , row.vars = c(3,2) ), 2)
```

```
                nodes     0     1     5    10
State size
noRel <=20 mm          0.02 -0.10  0.00  0.04
      >20-50 mm        0.18 -0.08 -0.07  0.06
      >50 mm           0.03 -0.13 -0.03  0.07
Rel   <=20 mm          0.02  0.07 -0.02 -0.02
      >20-50 mm       -0.05  0.00  0.03 -0.06
      >50 mm           0.04  0.10  0.06  0.02
Total <=20 mm          0.04 -0.03 -0.02  0.02
      >20-50 mm        0.13 -0.08 -0.04  0.00
      >50 mm           0.07 -0.03  0.03  0.09
```

```
> round(ftable((cX - cA) / cA*100, row.vars = c(3,2) ), 1)
                nodes    0    1    5   10
State size
noRel <=20 mm          0.2 -1.0  0.0  0.6
      >20-50 mm        2.1 -1.0 -1.1  1.3
      >50 mm           0.4 -1.8 -0.6  1.7
Rel   <=20 mm          1.1  3.4 -0.9 -0.8
      >20-50 mm       -2.1  0.1  1.5 -2.6
      >50 mm           1.8  4.6  2.7  1.0
Total <=20 mm          0.3 -0.3 -0.2  0.2
      >20-50 mm        1.2 -0.7 -0.4  0.0
      >50 mm           0.7 -0.4  0.3  1.5
```

The differences in the years spent in no relapse should be the same, thus the differences seen there should be purely simulation error.

Note that if we had a simulation-based *sample* of the probabilities as outlined above, we would be able to put confidence limits on the entries in this table as well.

The numbers in the tables above correspond to points at 15 years on the curves of "length of stay" in the C & L's SiM paper, so we could have generated these curves by using the cumulative sums instead, and the differences and ratios would then have been operations inside the resulting arrays.

Again, confidence intervals would be easiest to compute by using simulated data sets from many bootstrap samples, which are not implemented yet.

8 Metastases

A further state, "metastases" is recorded too. We included these among the relapses—relapse without metastases is at time `tor`, whereas metastases is at `tom`, regardless of previous relapse.

If we are willing to dispense with subdividing the deaths by the state from which they occurred we can split the original follow-up (in the `Lexis` object `Lbc`) in one go, using the `mcutLexis` function. Note that this requires that relapse dates recorded as equal to the metastasis dates be coded as `NA` thus treating relapse and metastasis as separate events (that can not occur at the same time). This is what we did when grooming the data initially, so we can cut the original `Lexis` object:

```
> mbc <- mcutLexis(Lbc,
+             timescale = "tfd",
+                    wh = c("tor", "tom"),
+            new.states = c("Rel", "Met"),
+            seq.states = TRUE,
+            new.scales = c("tfr", "tfm"))
NOTE: Precursor states set to Alive
> summary( mbc, timeScale = TRUE )
Transitions:
     To
From       Alive Dead Rel Rel-Met  Met  Records:  Events: Risk time:  Persons:
  Alive     1269  195 474       0 1044      2982     1713   17203.80      2982
  Rel          0   30 210     234    0       474      264    1436.23       474
  Rel-Met      0  187   0      47    0       234      187     485.92       234
  Met          0  860   0       0  184      1044      860    2144.79      1044
  Sum       1269 1272 684     281 1228      4734     3024   21270.74      2982

Timescales:
  tfd     A     P   tfr   tfm
   ""    ""    "" "Rel" "Met"
```

```
> mbc <- Relevel(mbc, list("Alive",
+                          "Rel",
+                    Met = c("Met","Rel-Met"),
+                          "Dead"))
> summary(mbc)

Transitions:
     To
From     Alive Rel  Met Dead  Records:  Events: Risk time:  Persons:
  Alive   1269 474 1044  195      2982      1713   17203.80      2982
  Rel        0 210  234   30       474       264    1436.23       474
  Met        0   0  231 1047      1278      1047    2630.71      1278
  Sum     1269 684 1509 1272      4734      3024   21270.74      2982

> # or: mbc <- Relevel(mbc, list(1, 3, Met = 4:5, 2))
> print(subset(mbc, lex.id %in% (1328+0:2))[,1:10],
+       row.names = FALSE, digits = 4)

       tfr tfm   tfd     A    P lex.dur lex.Cst lex.Xst lex.id  pid
        NA  NA 0.000 83.06 1985  1.8727   Alive     Rel   1329 1329
 2.220e-16  NA 1.873 84.93 1987  3.1923     Rel    Dead   1329 1329
        NA  NA 0.000 44.53 1994  2.4066   Alive     Rel   1328 1328
 0.000e+00  NA 2.407 46.93 1996  0.9254     Rel     Met   1328 1328
 9.254e-01   0 3.332 47.86 1997  4.0986     Met     Met   1328 1328
        NA  NA 0.000 68.92 1988  0.9090   Alive     Rel   1330 1330
        NA  NA 0.909 69.83 1988  1.0103     Rel     Met   1330 1330
 1.010e+00   0 1.919 70.84 1989  0.5530     Met    Dead   1330 1330
```

The lack of subdivision of deaths by state immediately preceding death can of course be remedied "by hand":

```
> xbc <- transform(mbc,
+                  lex.Xst = factor(ifelse(lex.Cst != "Alive" &
+                                          lex.Xst == "Dead",
+                                          paste("D(", lex.Cst, ")", sep=""),
+                                          as.character(lex.Xst))))
> xbc <- factorize( xbc )
NOTE: lex.Cst and lex.Xst now have levels:
 Alive Rel Met Dead D(Met) D(Rel)

> levels(xbc)

[1] "Alive"  "Rel"    "Met"    "Dead"   "D(Met)" "D(Rel)"

> xbc <- Relevel(xbc, c(1:4,6,5))
> levels(xbc)

[1] "Alive"  "Rel"    "Met"    "Dead"   "D(Rel)" "D(Met)"

> summary(xbc)

Transitions:
     To
From     Alive Rel  Met Dead D(Rel) D(Met)  Records:  Events: Risk time:  Persons:
  Alive   1269 474 1044  195      0      0      2982      1713   17203.80      2982
  Rel        0 210  234    0     30      0       474       264    1436.23       474
  Met        0   0  231    0      0   1047      1278      1047    2630.71      1278
  Sum     1269 684 1509  195     30   1047      4734      3024   21270.74      2982


> boxes(xbc, boxpos = list(x = c(15,40,15,85,85,85),
+                          y = c(85,50,15,85,50,15)),
+            show.BE = "nz",
+            scale.R = 100,
+                cex = 1.1)
```

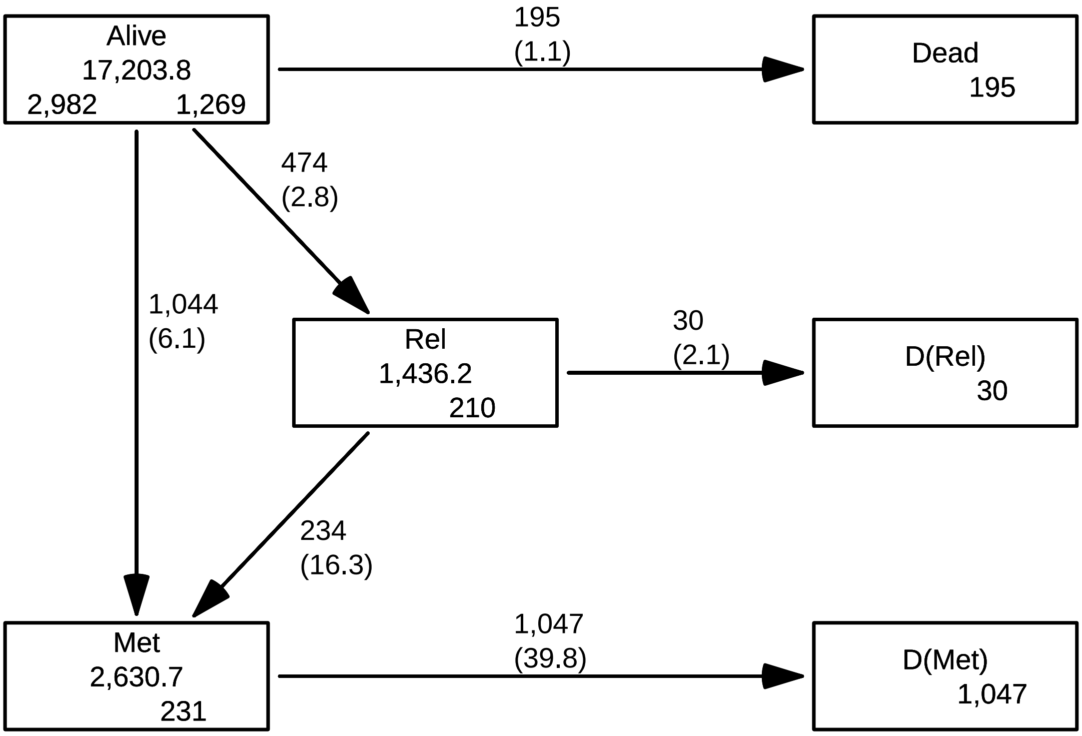

Figure 9: *Transitions when metastases are taken into account.* `./bcMS-box-rmx`

We could model all 6 transitions, exploring the possible effects of time since entry to the relapse and metastasis states as well as possible interactions. We might even model mortality rates from relapse and metastasis with some common parameters.

Eventually we would have specified some model for each of the transitions, and we could repeat the exercise from above, simulating state occupancies and time spent in different states.

So far this is left as an exercise to the reader. . .

References

[1] M. J. Crowther and P. C. Lambert. Parametric multistate survival models: Flexible modelling allowing transition-specific distributions with application to estimating clinically useful measures of effect differences. *Stat Med*, 36(29):4719–4742, Dec 2017.

9 What is still missing

The arrays `prA` and `prX` contain the probabilities of being in each of the four states (well, cumulated over states) as a function of time. Additionally, there are two more dimensions to the arrays corresponding to 5×3 combinations of two covariates (nodes and size) whereas other covariates (age, progesterone and hormone therapy) are fixed.

If we wanted some sort of uncertainty associated with the estimates we cold either simulate using repeat samples from the "posterior" distribution of the model parameters, or we could do a bootstrap of the original sample, re-estimating the models.

In terms of the simulated cohort, we would instead end up with, say 1000 cohorts, each of 100 people, and a corresponding extra dimension of 1000 on the arrays of probabilities. The could then be used for computation of confidence intervals for *any* type of measure we were to derive from the simulated cohorts.

Essentially measures of uncertainty would be referring to quantiles of the simulated probabilities (well, empirical fractions) from each of the samples of say 100, persons. Since each sample is devised to represent a probability we should take the sampling uncertainty into account when devising probabilities — that is not just use the empirical fractions but replace them by a sample from the posterior distribution of the probability given the empirical fraction.

If we use a flat prior for the probability, the posterior distribution of the probability given an observed fraction of x/n is Beta with shape $(x+1, n-x+1)$. Thus a simple deterministic jitter of the array of probabilities applied before computing the confidence limits. However, this does not take the time-dependence of the probabilities into account.

To be continued . . .

9.1 Technical note on `simLexis` implementation

The transition objects are large and clumsy, and may even contain the same models more than once. It would be better to only have the contents as the *names* of the transition models, and inside use `get` to construct the objects currently used:

```
> Tr <- lapply( Tr, lapply, get )
```

This will also make it easier to use bootstrapped data for evaluation of uncertainty. For a given bootstrap sample of data we would make updated model objects with names appended with some string, so that the input for each cycle of the simulation loop over bootstrap samples of data would be using an input transition object of the form:

```
> bootTr <- lapply( Tr, lapply, function(x) paste("BOOT",x,sep="") )
```

Generation of the model objects with these names would be using only the unique elements, avoiding fitting the same model more than once:

```
> unique.models <- unique( unlist( Tr ) )
> for( m in unique.models )
+    {
+    assign( paste("BOOT",m,sep=""),
+            update( get(m), data=boot.Lexis(data) ) )
+    }
```