Multistate example from Crowther & Lambert — with multiple timescales

SDCC
http://bendixcarstensen.com/AdvCoh
February 2018
Version 5

Compiled Monday 5th February, 2018, 22:26
from: /home/bendix/stat/R/lib.src/Epi/pkg/vignettes/BrCaMS.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 are available as the data set `BrCa` in the `Epi` package in a slightly modified form, where dates of relapse, metastasis and death are only non-`NA` for those that actually do see the events.

First we load the relevant packages and then the example data from the `Epi` package:

```
> library( Epi )
> library( popEpi )
> load( file="../data/BrCa.rda" )
> # data( BrCa )
> head( BrCa )
```

```
   pid year age meno      size grade nodes   pr    pr.tr  er hormon chemo tor tom      tod
1 1264 1986  54 post    <=20 mm     2     0 1360 7.215975 149     no    no  NA  NA       NA
2 1150 1990  55 post >20-50 mm     2     0  763 6.638568 763     no    no  NA  NA       NA
3  838 1988  34  pre    <=20 mm     2     0  113 4.736198 109     no    no  NA  NA       NA
4 1214 1990  42 post    <=20 mm     2     0  465 6.144186  79     no    no  NA  NA       NA
5 1130 1989  35  pre    <=20 mm     2     0   82 4.418841  25     no    no  NA  NA       NA
6 1118 1987  50 post    <=20 mm     2     0   75 4.330733  10     no    no  NA  NA 10.91855
        tox   xst
1 12.971937 Alive
2  8.783025 Alive
3  9.412731 Alive
4 10.472279 Alive
5 10.351814 Alive
6 10.918549  Dead
```

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"     "year"
[10] "age"     "meno"    "size"    "grade"   "nodes"   "pr"      "pr.tr"   "er"      "hormon"
[19] "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",
+     precursor.states = "Alive",
+            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:
  time.scale time.since
1        tfd
2          A
3          P
4        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.

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, )
```

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
  3.106   0.121   3.226
> 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
```

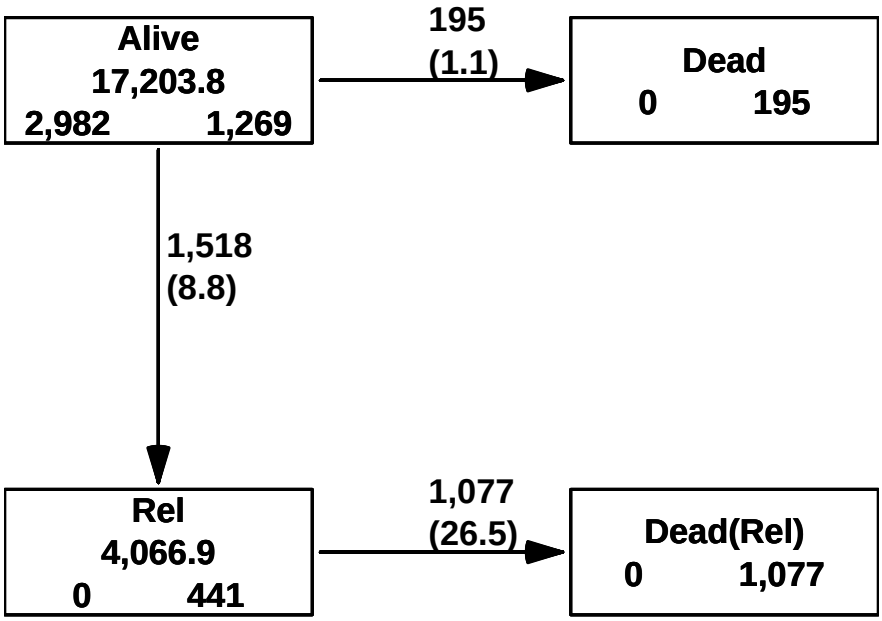

Figure 1: *Transitions in the correctly set up multistate model for the breast cancer survival dataset. Numbers* in *the boxes are person-years and (at the bottom) the number of persons starting resp. ending their follow-up in each state. Numbers on the arrows are the number of transitions and transition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

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

```
> system.time(
+ Sbc <- splitMulti( Rbc, tfd=seq(0,100,1/12) ) )
   user  system elapsed
  2.367   0.180   2.490
> 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?

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 )
> round( ftable( xtabs( cbind(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
lex.Fail Alive->Rel                0  1518     0         0
         Alive->Dead               0     0   195         0
         Rel->Dead(Rel)            0     0     0      1077
lex.dur  Alive->Rel            17133    63     8         0
         Alive->Dead           17133    63     8         0
         Rel->Dead(Rel)            0  4023     0        43
```

However, stacking data is needed only 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 variable 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( (lex.Xst=="Dead") ~ Ns( tfd, knots=kd.ad ),
+              offset = log( lex.dur ),
+              family = poisson,
+                data = subset( Sbc, lex.Cst=="Alive" ) )
> m.ar <- glm( (lex.Xst=="Rel") ~ Ns( tfd, knots=kd.ar ),
+              offset = log( lex.dur ),
+              family = poisson,
+                data = subset( Sbc, lex.Cst=="Alive" ) )
> m.rd <- glm( (lex.Xst=="Dead(Rel)") ~ Ns( tfd, knots=kd.rd ),
+              offset = log( lex.dur ),
+              family = poisson,
+                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: (lex.Xst == "Dead(Rel)") ~ Ns(tfd, knots = kd.rd)
Model 2: (lex.Xst == "Dead(Rel)") ~ Ns(tfd, knots = kd.rd) + Ns(tfr, knots = kr.rd)
Model 3: (lex.Xst == "Dead(Rel)") ~ 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 effect of `tfd` is clearly neither a feasible option with a deviance difference of 198 on 3 df. We shall deal with this extension later.

First we turn to the transition rates as function of time since diagnosis. Note that since the `lex.dur` is in units of PY, setting the value of it (as a covariate) to 100, means that we get the rates in units of 100 PY — basically rates in % per year:

```
> nd <- data.frame( tfd = seq(0,15,0.1),
+                   lex.dur = 100 )
> ad.rate <- ci.pred( m.ad, nd )
> ar.rate <- ci.pred( m.ar, nd )
> rd.rate <- ci.pred( m.rd, nd )
```

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

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

From the graph in figure 2 we see that the occurrence of relapse almost doubles over the first two years and then decreases. We also observe that the mortality RR between persons

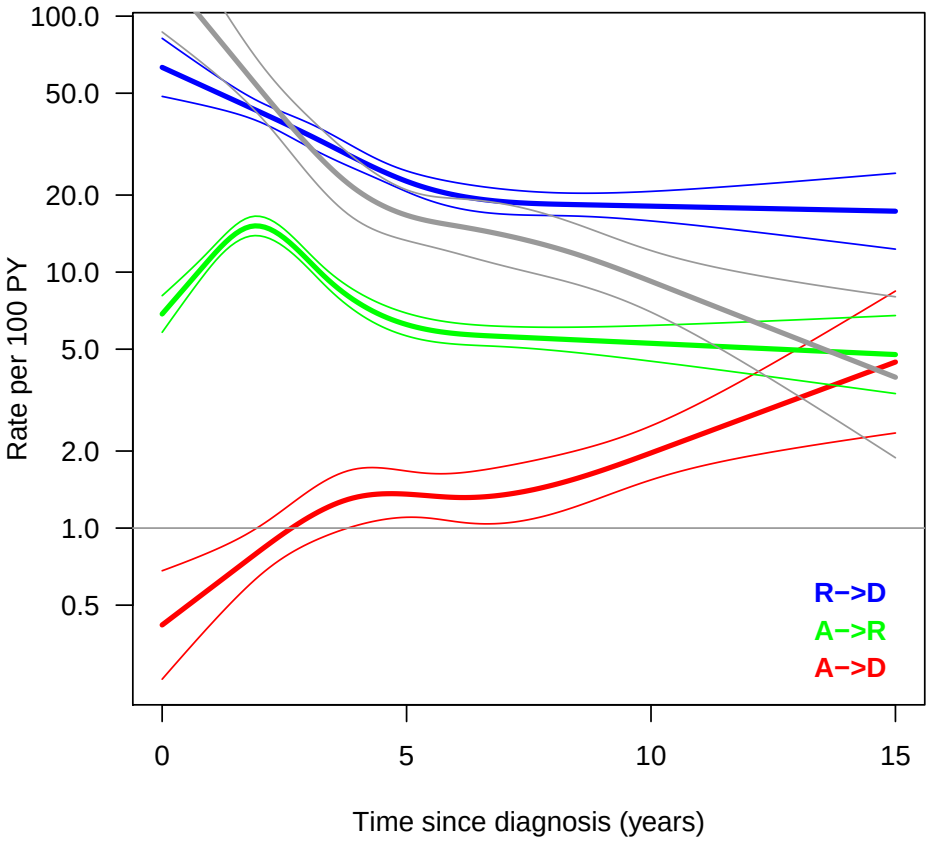

Figure 2: *Transition rates as function of time since diagnosis, the gray line is the mortality rate-ratio between persons with and without relapse — it seems as if the earlier the relapse, the higher the impact on mortality.*

with relapse and those without decreases from extremely high to about 5, a combination of decreasing mortality among persons with relapse and an increasing mortality among persons without relapse.

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`:

```
> 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: (lex.Xst == "Dead(Rel)") ~ Ns(tfd, knots = kd.rd)
Model 2: (lex.Xst == "Dead(Rel)") ~ Ns(tfd, knots = kd.rd) + Ns(tfr, knots = kr.rd)
Model 3: (lex.Xst == "Dead(Rel)") ~ 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 formally statistically significant effect of time at relapse, but the deviance change is much smaller than for the two timescales.

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 dpend 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 dispays 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.

```
> nd <- data.frame( expand.grid( tfd=c(NA,seq(0,15,0.1)),
+                                tad=c(0,0.5,1,2,3,5,8) ),
+                   lex.dur=100 )
> nd <- subset( transform( nd, tfr = tfd - tad ), tfr>=0 | is.na(tfr) )
> head( nd )
  tfd tad lex.dur tfr
1  NA   0     100  NA
2 0.0   0     100 0.0
3 0.1   0     100 0.1
4 0.2   0     100 0.2
5 0.3   0     100 0.3
6 0.4   0     100 0.4
> matplot( nd$tfd, cbind( ci.pred( x.rd, nd )[,1],
+                         ci.pred(xx.rd, nd )[,1] ),
+          type="l", lty=c("solid","22"), lend="butt",
+          lwd=3, col=clr[3], las=1,
+          log="y", xlab="Time since diagnosis (years)",
+                   ylab="Mortality rate per 100 PY" )
> matlines( seq(0,15,0.1), rd.rate,
+           type="l", lwd=c(3,1,1), lty=1, col=gray(0.3) )
```

From figure 3 we see that the simple model completely misses to describe the initial increase in mortality, and that the model without the time *at* relapse overestimates the mortality among women with early relapse.

4 Including covariates

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

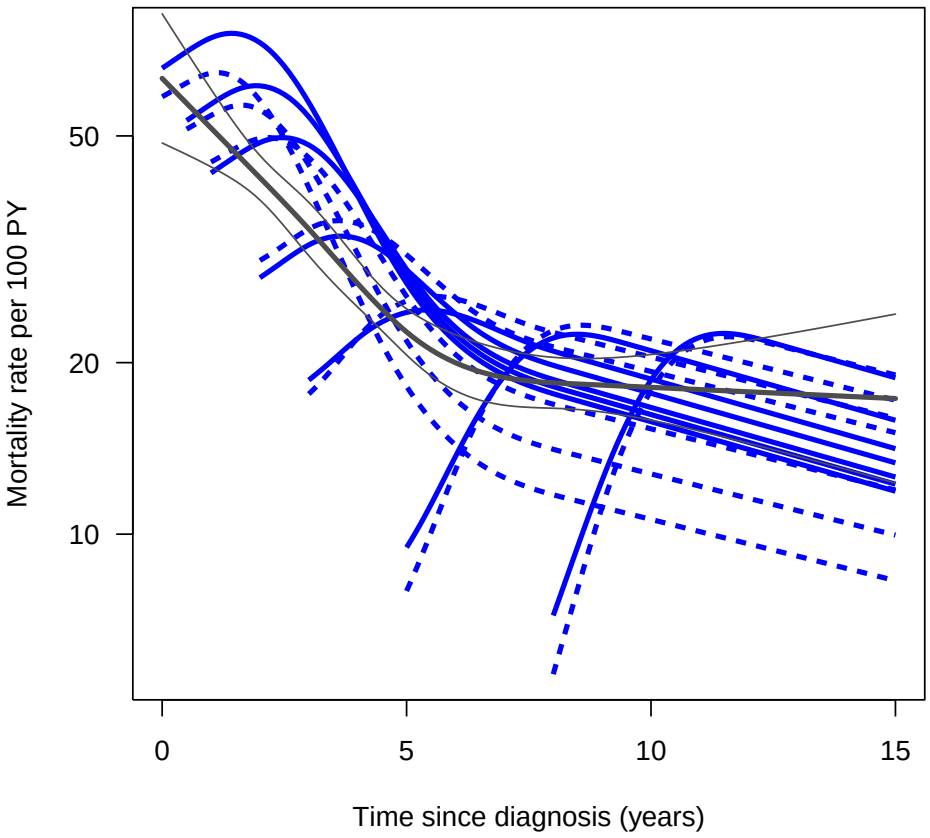

Figure 3: *Estimated mortality among women in relapse. The blue lines represent mortality for women relapsed at 0, 0.5, 1, 2, 3, 5, 8 years after diagnosis. The broken lines are predictions from the model where the time* at *relapse is modeled too. The gray line is from the model where only time since diagnosis is included ("proportional hazards model"), corresponding to the blue line in figure 2.*

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

4.1 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])
> 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="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   6.33
> 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 0.0421
> 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      2
```

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.

4.2 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 you should 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( xx.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 -22.6097 0.0000 -14.7533 -12.3995
Ns(tfd, knots = kd.ad)1  0.2873 0.2608   1.1020 0.2705  -0.2237   0.7984
Ns(tfd, knots = kd.ad)2  1.9852 0.2804   7.0811 0.0000   1.4357   2.5347
Ns(tfd, knots = kd.ad)3  1.1706 0.1944   6.0216 0.0000   0.7896   1.5516
age                      0.1286 0.0081  15.8762 0.0000   0.1128   0.1445
size>20-50 mm            0.1714 0.1610   1.0645 0.2871  -0.1442   0.4869
size>50 mm               0.4069 0.2330   1.7466 0.0807  -0.0497   0.8635
nodes                    0.0444 0.0184   2.4150 0.0157   0.0084   0.0804
pr.tr                    0.0305 0.0336   0.9069 0.3644  -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 -5.9670 0.0000 -1.2431 -0.6284
Ns(tfd, knots = kd.rd)1 -0.8855 0.1251 -7.0787 0.0000 -1.1306 -0.6403
Ns(tfd, knots = kd.rd)2 -1.3036 0.1670 -7.8080 0.0000 -1.6309 -0.9764
Ns(tfd, knots = kd.rd)3 -0.9527 0.1242 -7.6715 0.0000 -1.1961 -0.7093
age                     0.0049 0.0024  2.0240 0.0430  0.0002  0.0096
size>20-50 mm           0.1654 0.0712  2.3220 0.0202  0.0258  0.3050
size>50 mm              0.3266 0.0993  3.2892 0.0010  0.1320  0.5212
nodes                   0.0296 0.0058  5.1391 0.0000  0.0183  0.0409
pr.tr                  -0.2771 0.0396 -7.0016 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  4.7211 0.0000  0.0676  0.1635
```

```
> round( ci.lin( cx.rd ), 4 )
```

```
                              Estimate StdErr       z      P    2.5%   97.5%
(Intercept)                    -1.3261 0.1634 -8.1151 0.0000 -1.6464 -1.0058
Ns(tfd, knots = kd.rd)1        -1.2178 0.1394 -8.7367 0.0000 -1.4910 -0.9446
Ns(tfd, knots = kd.rd)2        -2.0109 0.2338 -8.6015 0.0000 -2.4691 -1.5527
Ns(tfd, knots = kd.rd)3        -0.9242 0.1443 -6.4032 0.0000 -1.2070 -0.6413
Ns(tfr, knots = kr.rd)1         0.9018 0.1327  6.7971 0.0000  0.6418  1.1619
Ns(tfr, knots = kr.rd)2         1.4849 0.2021  7.3468 0.0000  1.0887  1.8810
Ns(tfr, knots = kr.rd)3         0.6610 0.1313  5.0359 0.0000  0.4038  0.9183
Ns(tfd - tfr, knots = ka.rd)1   0.1422 0.0853  1.6667 0.0956 -0.0250  0.3094
Ns(tfd - tfr, knots = ka.rd)2   0.4578 0.1660  2.7579 0.0058  0.1324  0.7831
Ns(tfd - tfr, knots = ka.rd)3   0.0000 0.0000     NaN    NaN  0.0000  0.0000
age                             0.0048 0.0024  1.9830 0.0474  0.0001  0.0096
size>20-50 mm                   0.1449 0.0714  2.0308 0.0423  0.0051  0.2848
size>50 mm                      0.2914 0.0994  2.9306 0.0034  0.0965  0.4862
nodes                           0.0267 0.0057  4.6548 0.0000  0.0155  0.0380
pr.tr                          -0.1035 0.0139 -7.4251 0.0000 -0.1308 -0.0762
hormonyes                       0.1411 0.0972  1.4512 0.1467 -0.0495  0.3317
```

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.

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 ),
+                lex.dur = 100,
+                    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 ),
+                lex.dur = 100,
+                    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 ),
+                lex.dur = 100,
+                    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 ),
+          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 )
```

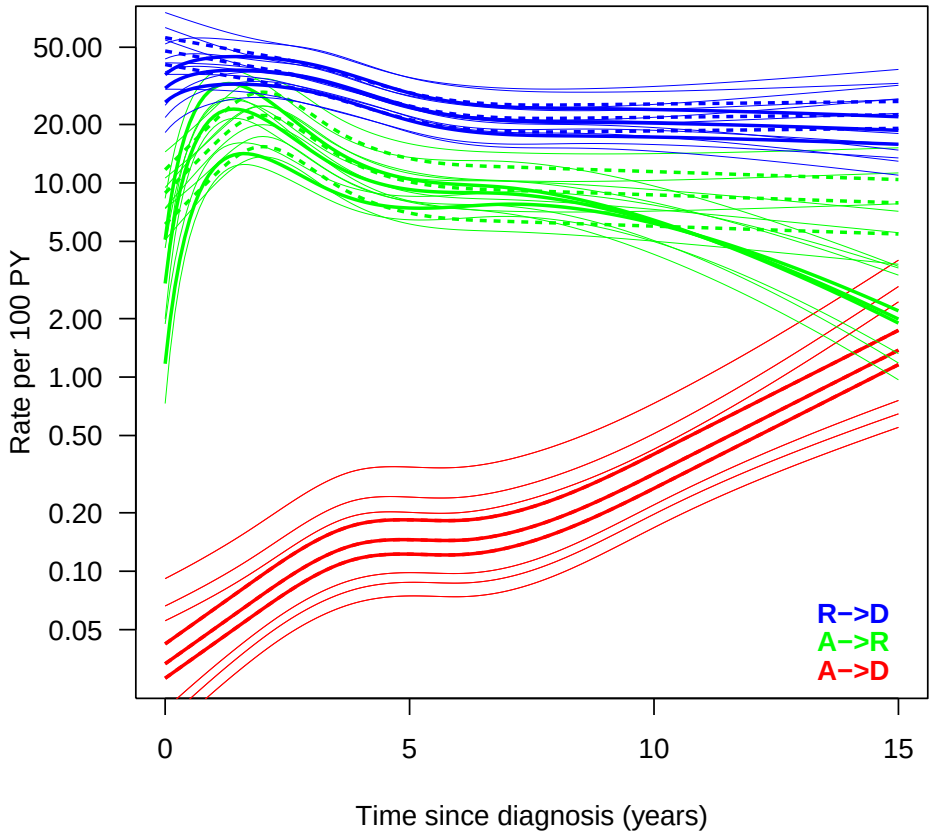

Figure 4: *Transition rates as function of time since diagnosis; the broken lines are from the main effects models and the full lines from the interaction model with* `age`*=54,* `nodes`*=5,* `pr.tr`*=3,* `hormon`*=no and where* `size` *assumes the values* < 20 *mm, 20–50 mm and* > 50 *mm (only the Alive→Rel transition). Thus the test of interaction is the comparison of the sets of parallel broken lines with the non-parallel full lines.*

```
> 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 ),
+          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 )
```

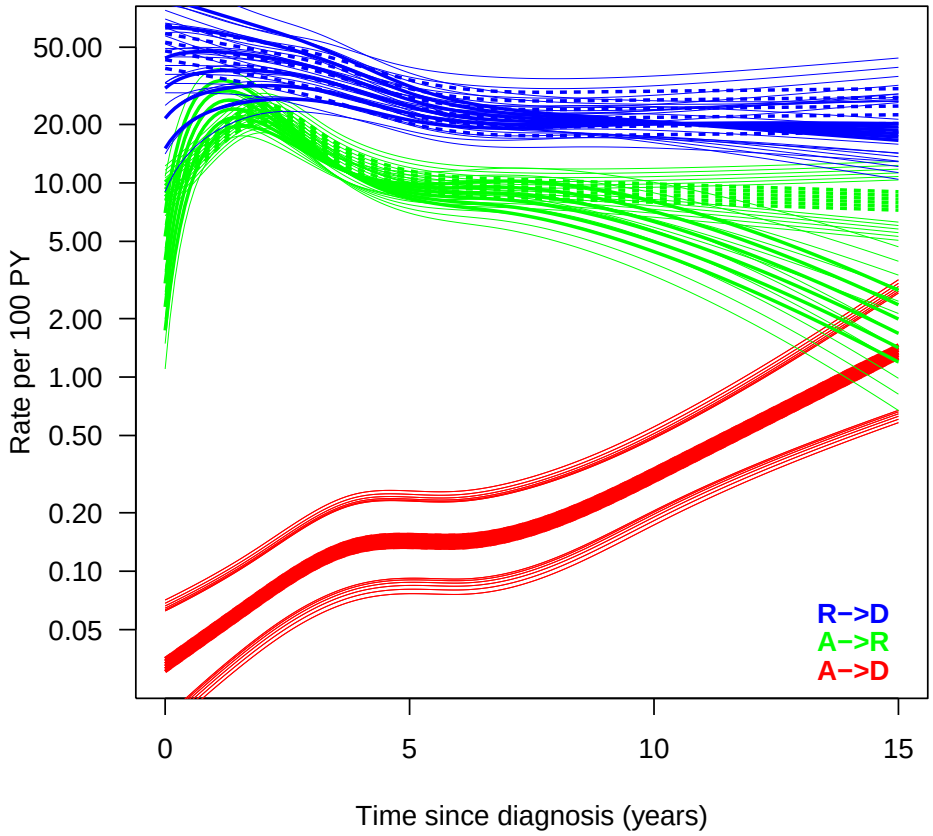

Figure 5: *Transition rates as function of time since diagnosis, the broken lines are from the main effects models and the full lines from the interaction model with* `age`*=54,* `size`*=20–50 mm,* `nodes`*=5,* `hormon`*=no and where* `pr.tr` *assumes the values 0–6. Thus the test of interaction is the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

```
> 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 ),
+          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 )
```

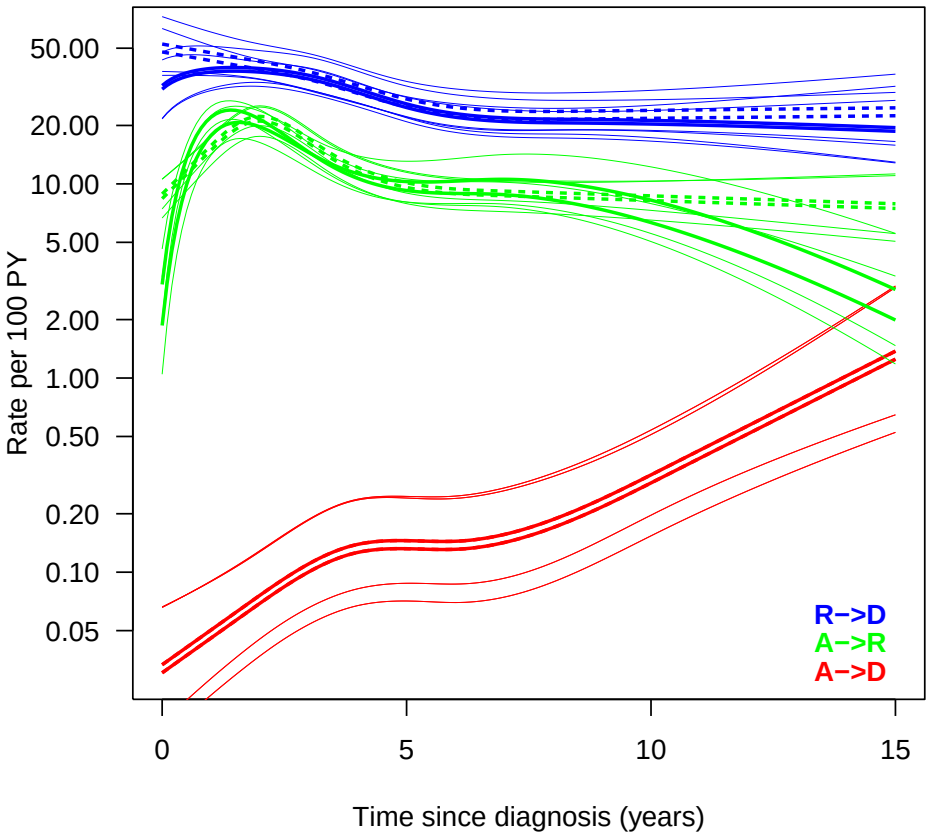

Figure 6: *Transition rates as function of time since diagnosis, the broken lines are from the main effects models and the full lines from the interaction model with* `age`*=54,* `size`*=20–50 mm,* `nodes`*=5,* `pr.tr`*=3 and where* `hormon` *assumes the values no and yes. Thus the test of interaction is the comparison of the sets of parallel broken lines with the non-parallel full lines.*

The general picture from the figures 4, 5 and 6 is the 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. If exploration of interactions were a major concern we might have used

5 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, 5, 10, 15 and 20 for this purpose.

5.1 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"  "pid"
[10] "year"    "age"     "meno"    "size"    "grade"   "nodes"   "pr"      "pr.tr"   "er"
[19] "hormon"  "chemo"   "tor"     "tom"     "tod"     "tox"     "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 <- seq(0,20,5)
> 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	5	3	no
5	0	NA	NA	NA	Alive	<NA>	NA	NA	54	>20-50 mm	5	3	no
6	0	NA	NA	NA	Alive	<NA>	NA	NA	54	>50 mm	5	3	no
7	0	NA	NA	NA	Alive	<NA>	NA	NA	54	<=20 mm	10	3	no
8	0	NA	NA	NA	Alive	<NA>	NA	NA	54	>20-50 mm	10	3	no
9	0	NA	NA	NA	Alive	<NA>	NA	NA	54	>50 mm	10	3	no
10	0	NA	NA	NA	Alive	<NA>	NA	NA	54	<=20 mm	15	3	no
11	0	NA	NA	NA	Alive	<NA>	NA	NA	54	>20-50 mm	15	3	no
12	0	NA	NA	NA	Alive	<NA>	NA	NA	54	>50 mm	15	3	no
13	0	NA	NA	NA	Alive	<NA>	NA	NA	54	<=20 mm	20	3	no
14	0	NA	NA	NA	Alive	<NA>	NA	NA	54	>20-50 mm	20	3	no
15	0	NA	NA	NA	Alive	<NA>	NA	NA	54	>50 mm	20	3	no

```
> str( Lini )
```

```
Classes 'Lexis' and 'data.frame':       15 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 5 5 5 10 10 10 15 ...
 $ 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
```

5.2 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:

```
> 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"
```

5.3 Simulation of a cohort

With this in place we can simulate:

```
> sL  <- simLexis( Tr=TR , init=Lini, N=2000, t.range=16 )
> sLx <- simLexis( Tr=TRx, init=Lini, N=2000, t.range=16 )
> save( sL, sLx, file="sL.Rda" )
```

We asked for simulation of 2000 persons with each of the 15 covariates patterns in `Lini`, a total of 30,000 persons:

```
> load( file="sL.Rda" )
> summary( sLx )
Transitions:
     To
From    Alive   Rel Dead Dead(Rel)  Records:  Events: Risk time:  Persons:
  Alive  4558 23989 1453         0     30000    25442  165921.78     30000
  Rel       0  1981    0     22008     23989    22008   79173.93     23989
  Sum    4558 25970 1453     22008     53989    47450  245095.71     30000
```

5.4 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 [1:161, 1:4] 1 1 1 1 0.997 ...
 - 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 15 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     2967      3146   3287
   5     3251      3429   3571
   10    3514      3719   3768
   15    3743      3858   3912
   20    3883      3962   3979
```

```
> prX <- prA <- NArray( c( dimnames( tt ), dimnames( pp ) ) )
> str( prA )
 logi [1:5, 1:3, 1:161, 1:4] NA NA NA NA NA NA ...
 - attr(*, "dimnames")=List of 4
  ..$ nodes: chr [1:5] "0" "5" "10" "15" ...
  ..$ 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="pr.Rda" )
> clr <- col2rgb( c("forestgreen","maroon") )
> clr <- cbind( clr, clr[,2:1]*0.6 + matrix(255,3,2)*0.4 )
> clr <- rgb( t(clr), max=255 )
> par( mfrow=c(3,3), mar=c(1,1.5,1,1), mgp=c(3,1,0)/1.6, oma=c(2,2,2,2) )
> nnn <- dimnames(prA)[[1]]
> sss <- dimnames(prA)[[2]]
> for( nn in nnn[c(1,3,5)] ) # only nodes as in the SiM paper
+ 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( "Size"  ,sss), side=3, at=c(1,3,5)/6, outer=TRUE, line=0, cex=0.66, las=0 )
> mtext( paste( "Nodes=",nnn[c(1,3,5)]), side=4, at=c(5,3,1)/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 )
```

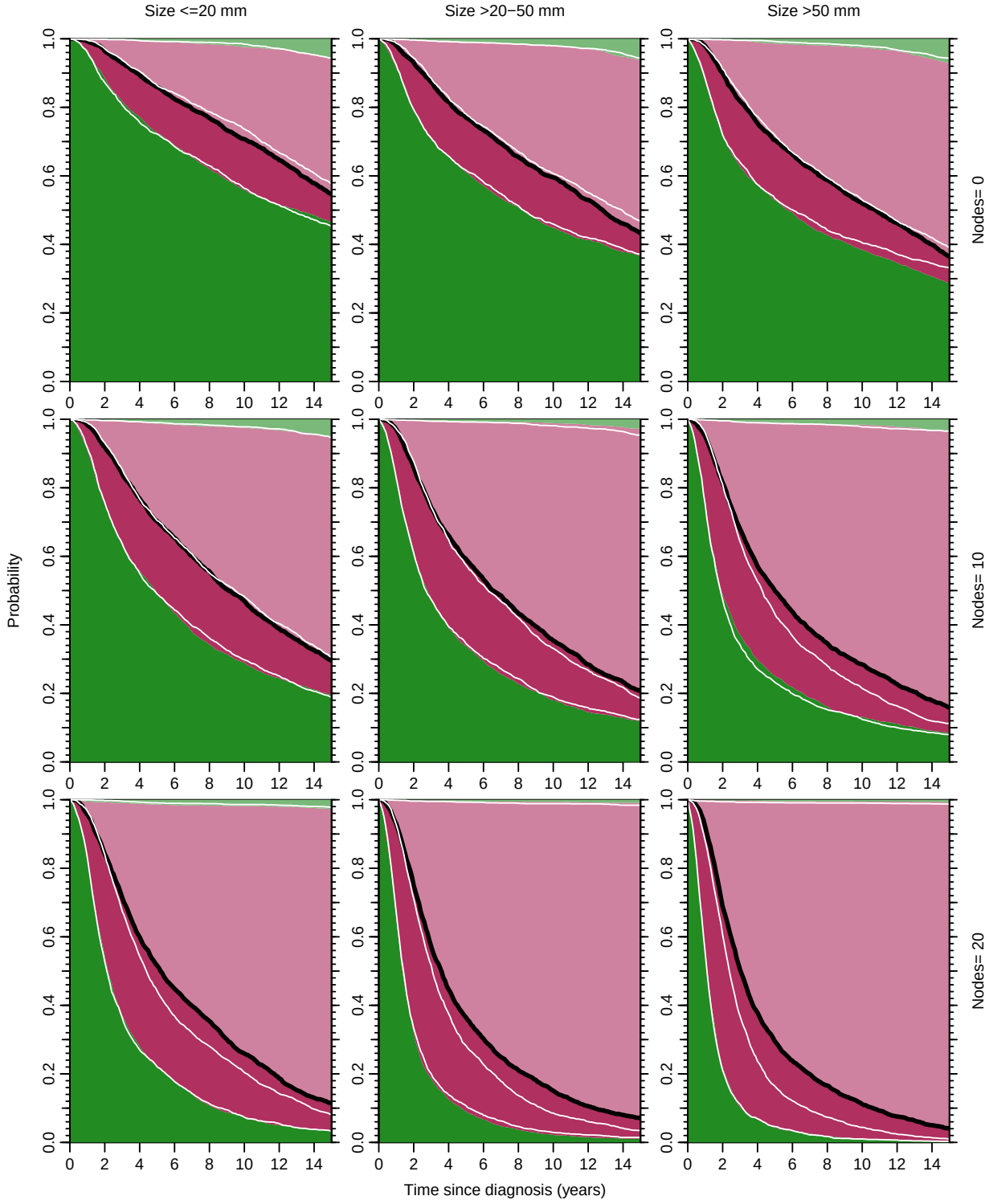

Figure 7: *Probabilities of being alive without relapse (green), with relapse (purple), dead after relapse (light purple), and dead without relapse (light green) The black line is the estimated survival curve. Computed from the model with effects of time since diagnosis as well as since relapse. The 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].*

6 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:

```
> cA <- apply( prA[,,1:151,1:3], c(1,2,4),
+              function(x) (sum(x[-1])+sum(x[-length(x)]))/2 * 1/10 )
> 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     5    10    15    20
State size
noRel <=20 mm          9.99  8.37  6.73  5.01  3.52
      >20-50 mm        8.61  6.98  5.04  3.16  2.28
      >50 mm           7.73  5.57  3.78  2.46  1.59
Rel   <=20 mm          2.03  2.38  2.54  2.66  2.44
      >20-50 mm        2.07  2.28  2.50  2.26  2.00
      >50 mm           2.01  2.22  2.18  1.88  1.66
Total <=20 mm         12.02 10.76  9.27  7.68  5.96
      >20-50 mm       10.68  9.26  7.53  5.42  4.29
      >50 mm           9.74  7.79  5.96  4.34  3.24

> cX <- apply( prX[,,1:151,1:3], c(1,2,4),
+              function(x) (sum(x[-1])+sum(x[-length(x)]))/20 )
> 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     5    10    15    20
State size
noRel <=20 mm         10.05  8.52  6.64  5.09  3.53
      >20-50 mm        8.52  6.84  4.95  3.27  2.13
      >50 mm           7.47  5.48  3.96  2.55  1.61
Rel   <=20 mm          1.72  2.16  2.50  2.98  3.11
      >20-50 mm        1.91  2.42  2.78  2.89  2.97
      >50 mm           2.09  2.56  2.76  2.82  2.84
Total <=20 mm         11.77 10.68  9.14  8.07  6.65
      >20-50 mm       10.42  9.26  7.72  6.16  5.10
      >50 mm           9.56  8.04  6.72  5.37  4.45
```

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.

Note that if we had a simulation-based sample of the probabilities as outlines 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 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 datasets from many bootstrap samples, which are not implemented yet.

7 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"),
+      precursor.states = "Alive",
+            new.states = c("Rel","Met"),
+            seq.states = TRUE,
+            new.scales = c("tfr","tfm") )
> 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:
  time.scale time.since
1        tfd
2          A
3          P
4        tfr        Rel
5        tfm        Met

> mbc <- Relevel( mbc, list( 1, 3, Met=4:5, 2 ) )

      type     old   new
1  lex.Cst   Alive Alive
2  lex.Cst    Dead
3  lex.Cst     Rel   Rel
4  lex.Cst Rel-Met   Met
5  lex.Cst     Met   Met
6  lex.Xst   Alive Alive
7  lex.Xst    Dead  Dead
8  lex.Xst     Rel   Rel
9  lex.Xst Rel-Met   Met
10 lex.Xst     Met   Met

> 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

> subset( mbc, lex.id %in% (1328+0:2) )[,1:10]
```

```
              tfr tfm       tfd        A        P   lex.dur lex.Cst lex.Xst lex.id  pid
1469           NA  NA 0.0000000 83.05832 1985.148 1.8726899   Alive     Rel   1329 1329
1470 2.220446e-16  NA 1.8726899 84.93101 1987.021 3.1923342     Rel    Dead   1329 1329
1942           NA  NA 0.0000000 44.52578 1993.908 2.4065709   Alive     Rel   1328 1328
1943 0.000000e+00  NA 2.4065709 46.93235 1996.315 0.9253936     Rel     Met   1328 1328
1944 9.253936e-01   0 3.3319645 47.85774 1997.240 4.0985622     Met     Met   1328 1328
1945           NA  NA 0.0000000 68.91837 1987.571 0.9089665   Alive     Rel   1330 1330
1946           NA  NA 0.9089665 69.82734 1988.480 1.0102670     Rel     Met   1330 1330
1947 1.010267e+00   0 1.9192335 70.83760 1989.490 0.5530457     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.Xst=="Dead" &
+                                                  lex.Cst!="Alive",
+                                                  paste( "D(",lex.Cst,")",sep=""),
+                                                  as.character(lex.Xst) ) ) )
> xbc <- Relevel( xbc )
> levels( xbc )
[1] "Alive"  "Rel"    "Met"    "Dead"   "D(Met)" "D(Rel)"

> xbc <- Relevel( xbc, c(1:3,5,4) )
> levels( xbc )
[1] "Alive"  "Rel"    "Met"    "D(Met)" "Dead"   "D(Rel)"

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


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

We could now 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.

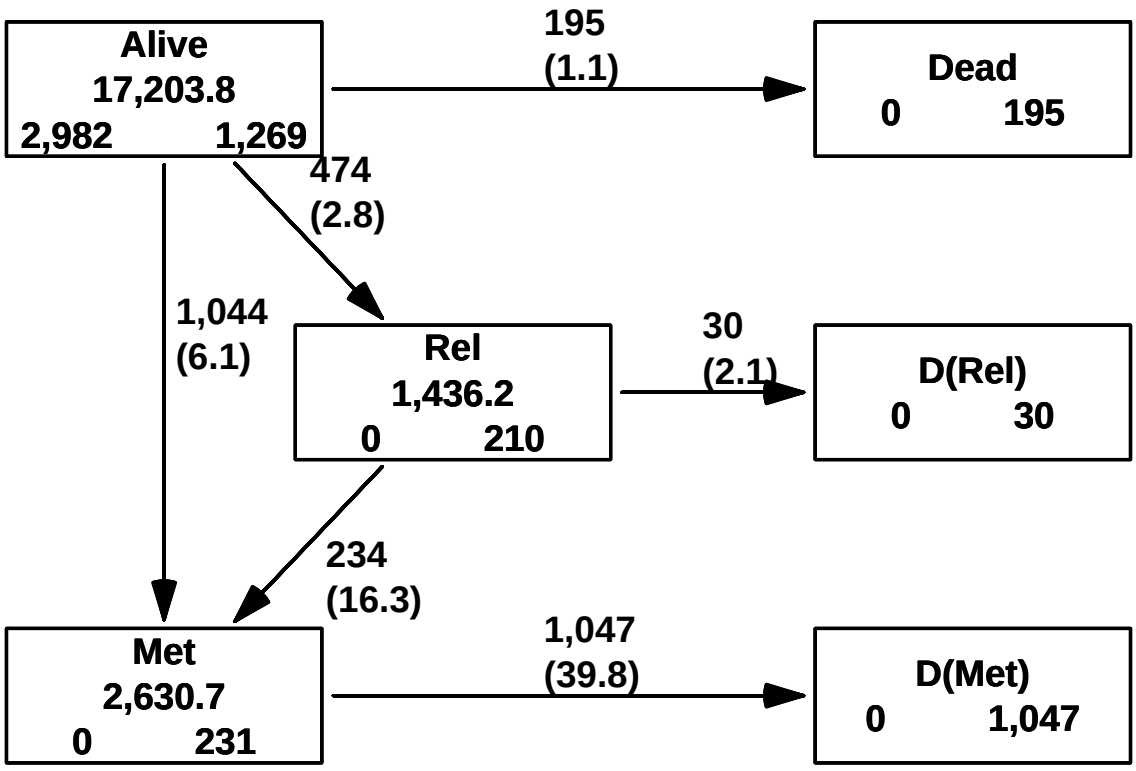

Figure 8: *Transitions when metastases are taken into account.*

8 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 ...

8.1 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) ) )
+     }
```