

Computing Cumulative Incidence Functions with the **etmCIF** Function, with a view Towards Pregnancy Applications

Arthur Allignol

1 Introduction

This paper documents the use of the **etmCIF** function to compute the cumulative incidence function (CIF) in pregnancy data.

2 Data Example

The data set **abortion**, included in the **etm** package will be used to illustrate the computation of the CIFs. We first load the **etm** package and the data set.

```
> require(etm)
> data(abortion)
```

Briefly, the data set contains information on 1186 pregnant women collected prospectively by the Teratology Information Service of Berlin, Germany (Meister and Schaefer, 2008). Among these pregnant women, 173 were exposed therapeutically to coumarin derivatives, a class of orally active anticoagulant, and 1013 women served as controls. Coumarin derivatives are suspected to increase the number of spontaneous abortions. Competing events are elective abortion (ETOP) and life birth.

Below is an excerpt of the data set

```
> head(abortion)

  id entry exit group cause
1  1     6   37     0     2
2  2     9   40     0     2
3  3    29   40     0     2
4  4    32   41     0     2
5  5    11   39     0     2
6  6    10   39     0     2
```

id is the individual number, **entry** is the gestational age at which the women entered the study, **exit** is the gestational age at the end of pregnancy, **group** is

the group membership (0 for controls and 1 for the women exposed to coumarin derivatives) and `cause` is the cause of end of pregnancy (1 for induced abortion, 2 for life birth and 3 for spontaneous abortion.)

3 Computing and plotting the CIFs

3.1 The `etmCIF` function

The CIFs are computed using the `etmCIF` function. It is a wrapper around the `etm` function, meant to facilitate the computation of the CIFs. `etmCIF` takes as arguments

- `formula`: A formula consisting of a `Surv` object on the left of a `~` operator, and the group covariate on the right. A `Surv` object is for example created this way: `Surv(entry, exit, cause != 0)`. We need to specify the entry time (`entry`), the gestational age at end of pregnancy (`exit`), and an event indicator (`cause != 0`). The latter means that any value different from 0 in `cause` will be considered as an event – which is the case in our example, as we don't have censoring.
- `data`: A data set in which to interpret the terms of the formula. In our case, it will be `abortion`.
- `etype`: Competing risks event indicator. When the status indicator is 1 (or TRUE) in the formula, `etype` describes the type of event, otherwise, for censored observation, the value of `etype` is ignored.
- `failcode`: Indicates the failure type of interest. Default is one. This option is only interesting for some features of the plot function.

3.2 Estimation and display of the CIFs

We know compute the CIFs

```
> cif.abortion <- etmCIF(Surv(entry, exit, cause != 0) ~ group,
+                       abortion, etype = cause, failcode = 3)
> cif.abortion
```

```
Call: etmCIF(formula = Surv(entry, exit, cause != 0) ~ group, data = abortion,
             etype = cause, failcode = 3)
```

```
Covariate: group
           levels: 0 1
```

```
group = 0
      time          P      se(P) n.event
CIF 1   43 0.04015931 0.009257784      20
```

```
CIF 2  43 0.79905931 0.022186468    924
CIF 3  43 0.16078139 0.021326113     69
```

```
group = 1
      time      P      se(P) n.event
CIF 1  42 0.2851118 0.04249308     38
CIF 2  42 0.3525651 0.04213898     92
CIF 3  42 0.3623231 0.04947340     43
```

Above is the display provided by the `print` function. It gives, at the last event time, the probabilities (P) standard errors (`se(P)`), and the total number of events (`n.event`) for the three possible pregnancy outcomes and for both groups.

More information is provided by the `summary` function.

```
> s.cif.ab <- summary(cif.abortion)
```

The function returns a list of data.frames that contain probabilities, variances, pointwise confidence intervals, number at risk and number of events for each event times. the `print` function displays this information for some selected event times.

```
> s.cif.ab
```

```
group=0

CIF 1
      P time      var      lower      upper n.risk n.event
0.00000000    4 0.000000e+00 0.00000000 0.00000000     18      0
0.03895488   13 8.444048e-05 0.02448808 0.06169378    645      1
0.04015931   23 8.570657e-05 0.02551262 0.06293875    819      0
0.04015931   33 8.570657e-05 0.02551262 0.06293875    885      0
0.04015931   39 8.570657e-05 0.02551262 0.06293875    716      0
0.04015931   43 8.570657e-05 0.02551262 0.06293875     6      0

CIF 2
      P time      var      lower      upper n.risk n.event
0.00000000    4 0.000000e+00 0.00000000 0.00000000     18      0
0.00000000   13 0.000000e+00 0.00000000 0.00000000    645      0
0.00000000   23 0.000000e+00 0.00000000 0.00000000    819      0
0.01354288   33 1.216141e-05 0.008169604 0.02241009    885      6
0.32201590   39 2.469805e-04 0.292313907 0.35391574    716    165
0.79905931   43 4.922394e-04 0.753968839 0.84061317     6      6

CIF 3
      P time      var      lower      upper n.risk n.event
0.0000000    4 0.0000000000 0.00000000 0.00000000     18      0
```

0.1504675	13	0.0004551293	0.1135970	0.1979013	645	1
0.1570366	23	0.0004553237	0.1199469	0.2041947	819	0
0.1599039	33	0.0004549788	0.1227389	0.2069247	885	0
0.1607814	39	0.0004548031	0.1235965	0.2077571	716	0
0.1607814	43	0.0004548031	0.1235965	0.2077571	6	0

group=1

CIF 1

	P time	var	lower	upper	n.risk	n.event
0.0000000	4	0.000000000	0.0000000	0.0000000	8	0
0.2604775	12	0.001785362	0.1879526	0.3542503	94	5
0.2811533	21	0.001802770	0.2074197	0.3742279	93	1
0.2851118	32	0.001805662	0.2111650	0.3780568	90	0
0.2851118	38	0.001805662	0.2111650	0.3780568	71	0
0.2851118	42	0.001805662	0.2111650	0.3780568	6	0

CIF 2

	P time	var	lower	upper	n.risk	n.event
0.0000000	4	0.000000e+00	0.000000000	0.000000000	8	0
0.0000000	12	0.000000e+00	0.000000000	0.000000000	94	0
0.0000000	21	0.000000e+00	0.000000000	0.000000000	93	0
0.01541036	32	6.016217e-05	0.005730041	0.04110307	90	1
0.13034947	38	5.576254e-04	0.090999609	0.18489993	71	12
0.35256510	42	1.775693e-03	0.276882073	0.44177471	6	6

CIF 3

	P time	var	lower	upper	n.risk	n.event
0.0000000	4	0.000000000	0.0000000	0.0000000	8	0
0.3143493	12	0.002571501	0.2266494	0.4254211	94	1
0.3507019	21	0.002482739	0.2627692	0.4576195	93	0
0.3584918	32	0.002459261	0.2706266	0.4644770	90	0
0.3584918	38	0.002459261	0.2706266	0.4644770	71	0
0.3623231	42	0.002447617	0.2744987	0.4678544	6	0

3.3 Plotting the CIFs

Interest lies in the CIFs of spontaneous abortion. We display them using the `plot` function, which by default, plots only the the CIFs for the event of interest, i.e., the one specified in `failcode`.

```
> plot(cif.abortion)
```

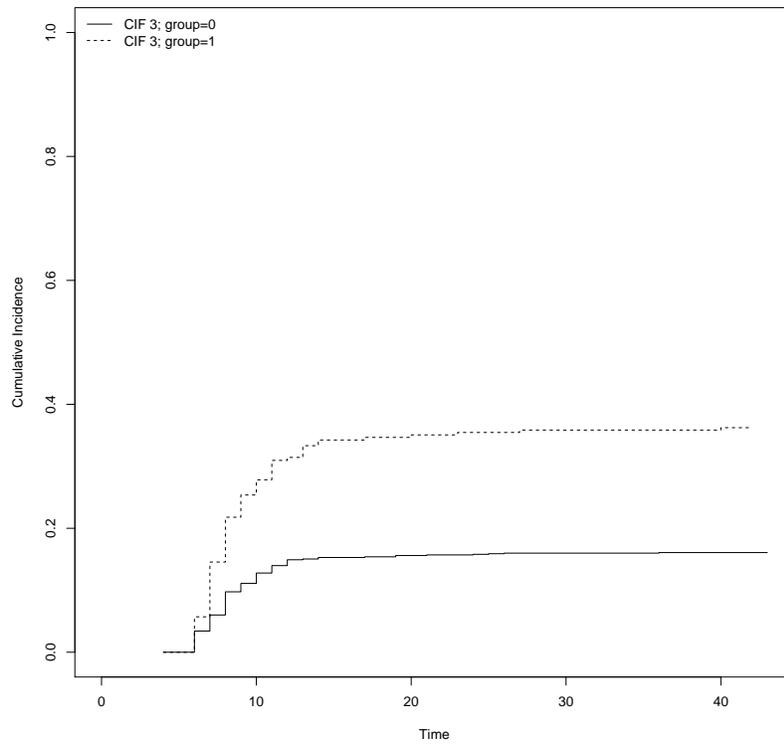


Figure 1: CIFs of spontaneous abortion for the controls (solid line) and the exposed (dashed line), using the default settings of the `plot` function.

We now add confidence intervals taken at week 27, plus a bit of customisation.

```
> plot(cif.abortion, curvlab = c("Control", "Exposed"), ylim = c(0, 0.6),  
+      ci.type = "bars", pos.ci = 27, col = c(1, 2), ci.lwd = 6,  
+      lwd = 2, lty = 1, cex = 1.3)
```

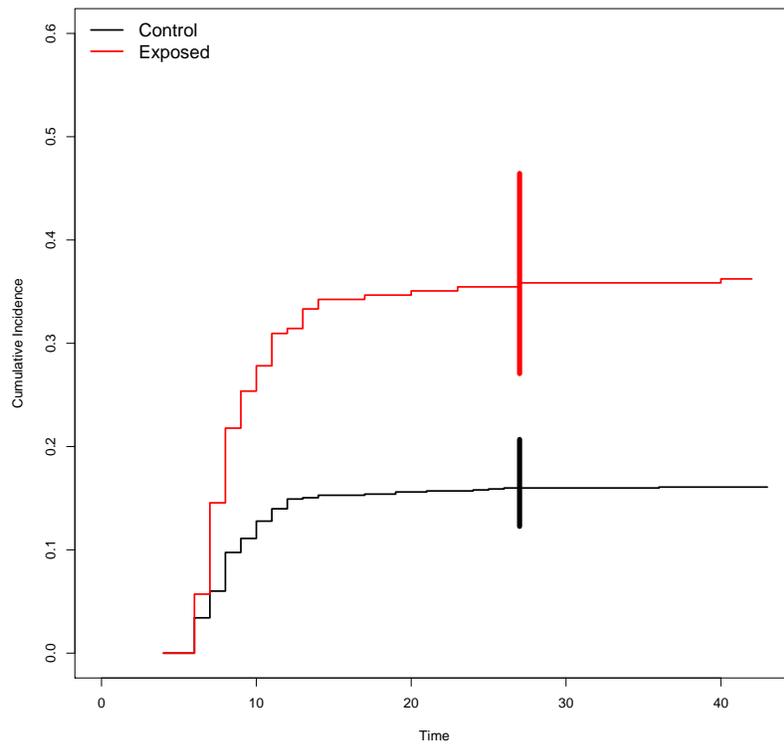


Figure 2: CIFs of spontaneous abortion for the controls (black) and the exposed (red), along with pointwise confidence intervals taken at week 27.

When the figure is to be in black and white, or when the confidence intervals are not as separated as in this example, it might be a good idea to shift slightly one of the bars representing the confidence interval, so that the two bars don't overlap. This might be done manipulating the `pos.ci` argument:

```
> plot(cif.abortion, curvlab = c("Control", "Exposed"), ylim = c(0, 0.6),  
+      ci.type = "bars", pos.ci = c(27, 28), col = c(1, 1), ci.lwd = 6,  
+      lwd = 2, lty = c(2, 1), cex = 1.3)
```

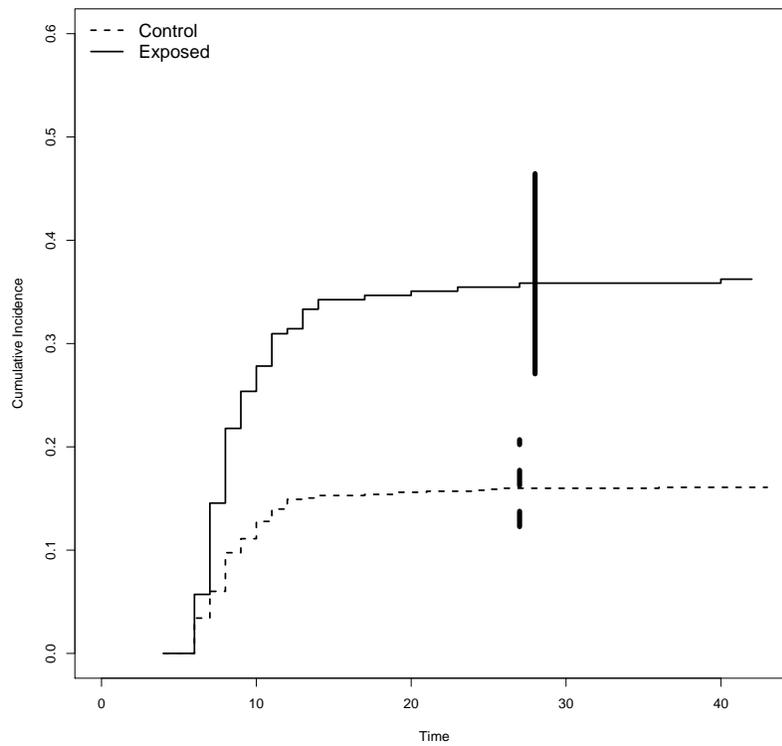


Figure 3: CIFs of spontaneous abortion for the controls (dashed line) and the exposed (solid line), along with pointwise confidence intervals.

Pointwise confidence intervals can also be plotted for the whole follow-up period.

```
> plot(cif.abortion, curvlab = c("Control", "Exposed"), ylim = c(0, 0.5),  
+      ci.type = "pointwise", col = c(1, 2), lwd = 2, lty = 1, cex = 1.3)
```

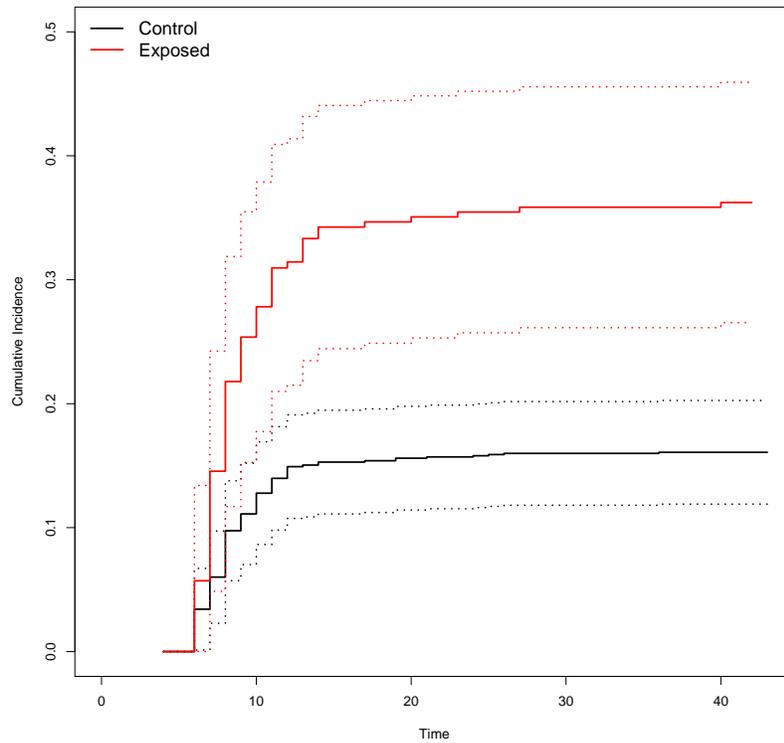


Figure 4: Same as the last pictures, except for the confidence intervals, that are displayed for the whole follow-up period.

CIFs for other pregnancy outcomes can also be plotted using the `which.cif` arguments. For instance, for plotting the CIFs of ETOP and life birth on the same graph, we specify `which.cif = c(1, 2)` in the call to `plot`.

```
> plot(cif.abortion, which.cif = c(1, 2), ylim = c(0, 0.8), lwd = 2,  
+      col = c(1, 1, 2, 2), lty = c(1, 2, 1, 2), legend = FALSE)  
> legend(0, 0.8, c("Control", "Exposed"), col = c(1, 2), lty = 1,  
+      bty = "n", lwd = 2)  
> legend(0, 0.7, c("ETOP", "Life Birth"), col = 1, lty = c(1, 2),  
+      bty = "n", lwd = 2)
```

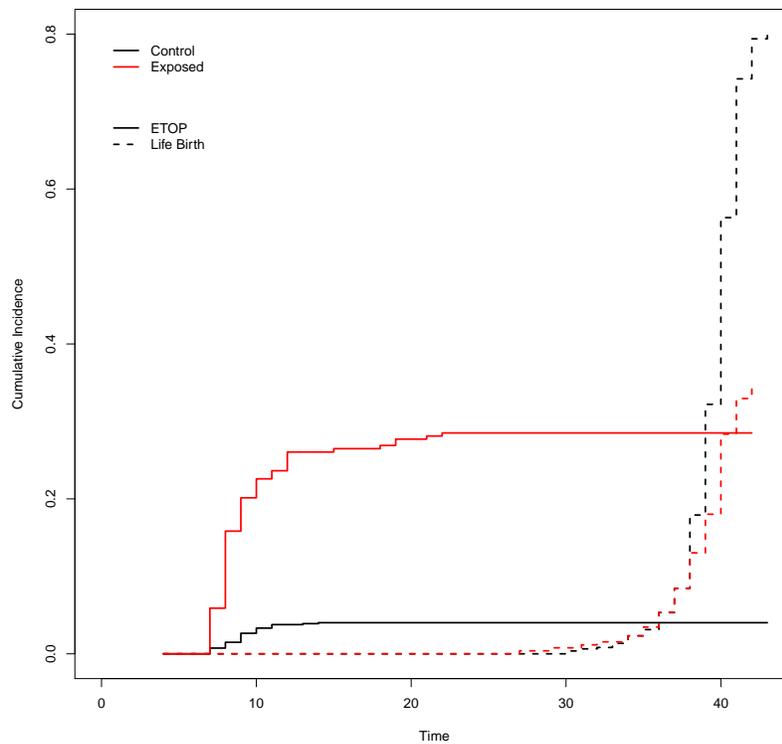


Figure 5: CIFs of ETOP (solid lines) and life birth (dashed lines) for the exposed, in red, and the controls, in black.

3.4 Some More Features

Competing event names For those who don't like using plain numbers for naming the competing events or the group allocation, it is of course possible to give more informative names, either as factors or character vectors. For instance, we define a new group variable that takes value 'control' or 'exposed', and we give more informative names for the pregnancy outcomes.

```
> abortion$status <- with(abortion, ifelse(cause == 2, "life birth",
+                                       ifelse(cause == 1, "ETOP", "spontaneous abortion")))
> abortion$status <- factor(abortion$status)
> abortion$treat <- with(abortion, ifelse(group == 0, "control", "exposed"))
> abortion$treat <- factor(abortion$treat)
```

We can compute the CIFs as before, taking care of changing the failcode argument.

```
> new.cif <- etmCIF(Surv(entry, exit, status != 0) ~ treat, abortion,
+                  etype = status, failcode = "spontaneous abortion")
> new.cif
```

```
Call: etmCIF(formula = Surv(entry, exit, status != 0) ~ treat, data = abortion,
             etype = status, failcode = "spontaneous abortion")
```

```
Covariate: treat
           levels: control exposed
```

```
treat = control
```

	time	P	se(P)	n.event
CIF ETOP	43	0.04015931	0.009257784	20
CIF life birth	43	0.79905931	0.022186468	924
CIF spontaneous abortion	43	0.16078139	0.021326113	69

```
treat = exposed
```

	time	P	se(P)	n.event
CIF ETOP	42	0.2851118	0.04249308	38
CIF life birth	42	0.3525651	0.04213898	92
CIF spontaneous abortion	42	0.3623231	0.04947340	43

The `summary` and `plot` functions will work as before, except for a more informative outcome from scratch.

Taking advantage of the miscellaneous functions defined for etm objects The `etmCIF` function uses the more general `etm` machinery for computing the CIFs. Thus the returned `etmCIF` object is for part a list of `etm` objects (one for each covariate level). It is therefore relatively easy to use the methods defined for `etm` on `etmCIF` objects.

An example would be to use the `trprob` function to extract the CIF of spontaneous abortion for the controls. This function takes as arguments an `etm` object, the transition we are interested in, in the form “from to” (the state a patient comes from is automatically defined as being 0 in `etmCIF`), and possibly some time points. Using `new.cif` from the example above:

```
> trprob(new.cif[[1]], "0 spontaneous abortion", c(1, 10, 27))  
[1] 0.0000000 0.1278037 0.1599039
```

We applied the `trprob` function to the `etm` object for the controls (which is in the first item of the output, for the exposed in the second). The transition of interest is from 0 to `spontaneous abortion`, and we want the CIF at weeks 1, 10 and 27 (just put nothing if you want the CIF for all time points).

Another example would be to use the `lines` function to add a CIF to an existing plot. The following code snippet adds the CIF of ETOP for the exposed to Figure 3. That’s the `tr.choice` arguments that defines which CIF to pick. It works in the same way as in the `trprob` function.

```
> lines(cif.abortion[[2]], tr.choice = "0 1", col = 2, lwd = 2)
```

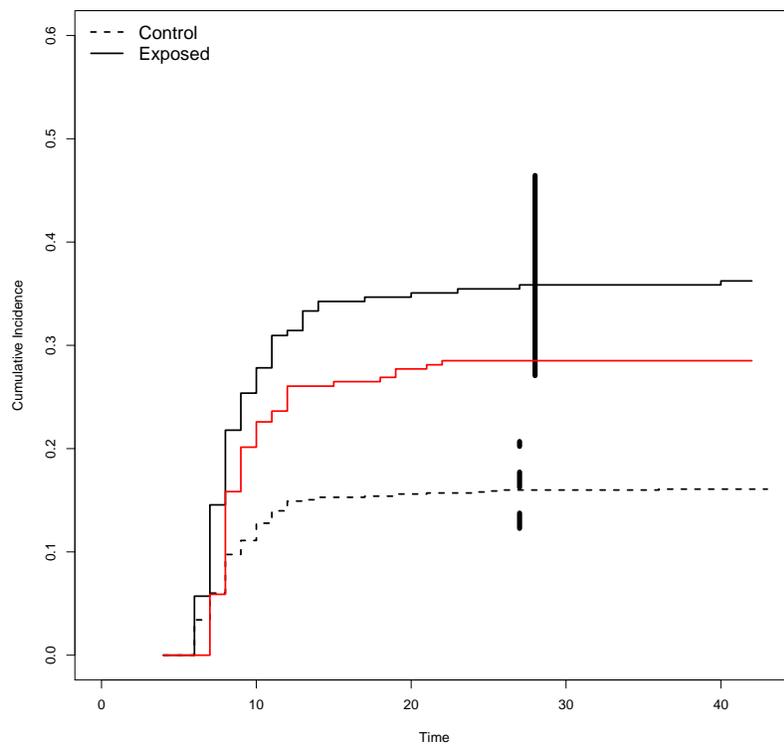


Figure 6: Figure 3 along with the CIF of ETOP for the exposed in red.

References

- Meister, R. and Schaefer, C. (2008). Statistical methods for estimating the probability of spontaneous abortion in observational studies—analyzing pregnancies exposed to coumarin derivatives. *Reproductive Toxicology*, 26(1):31–35.