`surrosurv`: an `R` Package for the Evaluation of Failure Time Surrogate Endpoints in Individual Patient Data Meta-Analyses of Randomized Clinical Trials

Federico Rotolo[1,2] Xavier Paoletti[1,2] Stefan Michiels[1,2]
[1] Service de Biostatistique et d'Épidémiologie, Gustave Roussy, Univ. Paris-Saclay
[2] INSERM U1018 OncoStat, CESP, Univ. Paris-Sud, Univ. Paris-Saclay

Abstract

Background and Objective. Surrogate endpoints are attractive for use in clinical trials instead of well-established endpoints because of practical convenience. To validate a surrogate endpoint, two important measures can be estimated in a meta-analytic context when individual patient data are available: the R^2_{indiv} or the Kendall's τ at the individual level, and the R^2_{trial} at the trial level. We aimed at providing an `R` implementation of classical and well-established as well as more recent statistical methods for surrogacy assessment with failure time endpoints. We also intended incorporating utilities for model checking and visualization and data generating methods described in the literature to date.

Methods. In the case of failure time endpoints, the classical approach is based on two steps. First, a Kendall's τ is estimated as measure of individual level surrogacy using a copula model. Then, the R^2_{trial} is computed via a linear regression of the estimated treatment effects; at this second step, the estimation uncertainty can be accounted for via measurement-error model or via weights. In addition to the classical approach, we recently developped an approach based on bivariate auxiliary Poisson models with individual random effects to measure the Kendall's τ and treatment-by-trial interactions to measure the R^2_{trial}. The most common data simulation models described in the literature are based on: copula models, mixed proportional hazard models, and mixture of half-normal and exponential random variables.

Results. The `R` package `surrosurv` implements the classical two-step method with Clayton, Plackett, and Hougaard copulas. It also allows to optionally adjust the second-step linear regression for measurement-error. The mixed Poisson approach is implemented with different reduced models in addition to the full model. We present the package functions for estimating the surrogacy models, for checking their convergence, for performing leave-one-trial-out cross-validation, and for plotting the results. We illustrate their use in practice on individual patient data from a meta-analysis of 4069 patients with advanced gastric cancer from 20 trials of chemotherapy.

Conclusions. The `surrosurv` package provides an `R` implementation of classical and recent statistical methods for surrogacy assessment of failure time endpoints. Flexible simulation functions are available to generate data according to the methods described in the literature.

1. Introduction

Surrogate endpoints are endpoints which can reliably be used instead of well-established (true) endpoints and which yield improved practical convenience in terms of lower cost, more rapid occurrence, increased ease of assessment, or reduced invasiveness [4]. Two conditions must be fulfilled for surrogate endpoint to be reliable: it must be strongly associatied with the true endpoint at the individual level and the effect of the treatment on it must be strongly assciated with the effect on the true endpoint. In a meta-analytic context and when the endpoints are gaussian [5], the usual measure of individual level surrogacy is the R^2_{indiv} between the *endpoints*, which measures the part of variability of the true endpoint T explained by the surrogate endpoint S. At the trial level, the usual measure of surrogacy is given by the R^2_{trial} between the *treatment effects* on the two endpoints, that measures the part of variability of the *treatment effect* on T explained by the *treatment effect* on S.

In the case of failure time (survival) endpoints, the classical methods developped for normally-distributed endpoints cannot be used because of right censoring. Burzykowski and colleagues [3] developped a meta-analytic model for failure time endpoints that measures individual level surrogacy in terms of Kendall's τ [18] and trial level surrogacy in terms of R^2_{trial}. This method is largely employed in numerous applications in the medical literature. Because of some limitations including convergence issues, the interpretation of the results is difficult in some cases [26, 2]. Recently, we considered using bivariate mixed proportional hazard models [10], which are the most natural adaptation of the above-mentioned meta-analytic approach by Buyse et al. [5] to the survival case. We exploited [36] the connection between the proportional hazard models and the Poisson log-linear models [40, 20] to build the joint model for the two treatment effects adjusted for individual dependence and baseline heterogeneity across trials.

In the present paper, we show how the classical and more recent models can be fitted by use of the `R` [29] package `surrosurv` [34]. Model checking can be performed thanks to utilities for convergence assessment and leave-one-trial-out crossvalidation. User-friendly functions allow the user to clearly show the results of the estimated models. We illustrate the available functions using individual data of a meta-analysis of 20 randomized trials of chemotherapy, including 4069 patients with advanced/recurrent gastric cancer [14, 27].

2. Computational methods and theory

Let T_{ij} and S_{ij} be the times to the true and the surrogate endpoints, respectively, for patient $j \in \{1, \ldots, n_i\}$ in trial $i \in \{1, \ldots, N\}$. Let Z_{ij} be the indicator of the treatment arm to which the j-th patient in the i-th trial has been randomized.

2.1. Two-step copula approach

The model proposed by Burzykowski et al. [3] for failure time endpoints consists in two steps, one for the individual and one for the trial level.

Individual-level. At the first step, the bivariate proportional hazard model is defined by means of the marginal hazard functions and of the copula function to account for their dependence:

$$\begin{cases} h_{Sij}(s; Z_{ij}) = h_{Si}(s) \exp\left\{\alpha_i Z_{ij}\right\} \\ h_{Tij}(t; Z_{ij}) = h_{Ti}(t) \exp\left\{\beta_i Z_{ij}\right\} \\ C_\theta(S_{Sij}(s), S_{Tij}(t)) \end{cases} \tag{1}$$

where $h_{Si}(s)$ and $h_{Ti}(s)$ are the trial-specific baseline hazards, α_i and β_i the treatment effects, and $S_{Sij}(s)$ and $S_{Tij}(t)$ the survival functions associated to h_{Tij} and h_{Tij}. The dependence parameter θ is reparametrized into the individual-level Kendall's τ, according to the copula function thanks to the `tau()` function in the `copula` package [16, 41].

In the `surrosurv` package, Weibull marginal hazards are implemented, together with three copula functions:

- the Clayton copula [7]

$$C_\theta(u, v) = \left(u^{-\theta} + v^{-\theta} - 1\right)^{-1/\theta}, \tag{2}$$

  with $\theta > 0$ and Kendall's $\tau = \theta/(\theta + 2)$;

- the Plackett copula [28]

$$\begin{aligned} C_\theta(u, v) &= \left[Q - R^{1/2}\right] / \left[2(\theta - 1)\right], \\ Q &= 1 + (\theta - 1)(u + v), \\ R &= Q^2 - 4\theta(\theta - 1)uv, \end{aligned} \tag{3}$$

  with $\theta > 0$ and Kendall's τ computed using numerical integration as no analytical expression is available;

- the Hougaard copula [17]

$$C_\theta(u, v) = \exp\left(-\left[(-\ln u)^{1/\theta} + (-\ln v)^{1/\theta}\right]^\theta\right), \tag{4}$$

  with $\theta \in (0, 1)$ and Kendall's $\tau = 1 - \theta$.

Further details on these three copula models can be found in the `vignette('copula', package = 'surrosurv')`.

Trial level. At the second step, the estimates of the treatment effects obtained at the first step are assumed to follow the mixed model

$$\begin{pmatrix} \hat{\alpha}_i \\ \hat{\beta}_i \end{pmatrix} = \begin{pmatrix} \alpha_i \\ \beta_i \end{pmatrix} + \begin{pmatrix} \epsilon_{ai} \\ \epsilon_{bi} \end{pmatrix}, \tag{5}$$

$$\begin{pmatrix} \alpha_i \\ \beta_i \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \boldsymbol{D} = \begin{pmatrix} d_a^2 & d_a d_b \rho_{\text{trial}} \\ d_a d_b \rho_{\text{trial}} & d_b^2 \end{pmatrix} \right), \tag{6}$$

$$\begin{pmatrix} \epsilon_{ai} \\ \epsilon_{bi} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \boldsymbol{\Omega}_i = \begin{pmatrix} \omega_{ai}^2 & \omega_{ai}\omega_{bi}\rho_{\epsilon i} \\ \omega_{ai}\omega_{bi}\rho_{\epsilon i} & \omega_{bi}^2 \end{pmatrix} \right). \tag{7}$$

where $(\alpha_i, \beta_i)'$ are the true treatment effects and $(\epsilon_{ai}, \epsilon_{bi})'$ the estimation errors.

The trial-level surrogacy measure is $R^2_{\text{trial}} = \rho^2_{\text{trial}}$. In practice, we compute the ρ_{trial} via a linear regression of the β_i's over the α_i's adjusted by measurement error by fixing the $\boldsymbol{\Omega}_i$'s at their estimates from the first step [39] by using the `mvmeta` package [12, 11]. This adjusted (for measurement error) model is sometimes computationally challenging and does not always converge. The `surrosurv` package returns also the so-called unadjusted R^2_{trial}, obtained using a linear regression — equivalent to fixing all the elements of $\boldsymbol{\Omega}_i$ equal to 0 — by weigthing the observations $(\alpha_i, \beta_i)'$ by the trial size, in order to account somehow indirectly and approximately for estimation uncertainty.

2.2. One-step mixed Poisson approach

Let us assume that the bivariate proportional hazard model given by the first two lines of equation (1) holds conditionally on an individual random effect $u_{ij} \sim \mathcal{N}(0, \sigma^2_{\text{indiv}})$:

$$\begin{cases} h_{Sij}(s \mid u_{ij}) = h_{Si}(s) \exp\{u_{ij} + \alpha_i Z_{ij}\} \\ h_{Tij}(t \mid u_{ij}) = h_{Ti}(t) \exp\{u_{ij} + \beta_i Z_{ij}\}. \end{cases} \tag{8}$$

Note that this corresponds to a shared frailty model with bivariate clusters [10]. The shared frailty term u_{ij} accounts for individual level dependence.

It is well-known (see for instance [40, 9]) that the parameters of Cox models can be estimated by fitting a so-called 'auxiliary' Poisson log-linear regression model, by dividing the time scale into intervals $k = 1, \ldots, K$. The auxiliary Poisson model provides the same estimator as the Cox model if the bounds of the intervals are all the observed event times, and an approximation of the Cox estimators otherwise. In the surrogacy assessment context, the parameters of the bivariate frailty model (8) can be estimated via a bivariate mixed Poisson model

$$\begin{cases} \log\left(\mu^{(k)}_{Sij}\right) = \mu^{(k)}_{Si} + u_{ij} + \alpha_i Z_{ij} + \log\left(y^{(k)}_{Sij}\right) \\ \log\left(\mu^{(k)}_{Tij}\right) = \mu^{(k)}_{Ti} + u_{ij} + \beta_i Z_{ij} + \log\left(y^{(k)}_{Tij}\right) \end{cases} \tag{9}$$

with $y_{Sj}^{(k)}$ and $y_{Tj}^{(k)}$ the time spent at risk by subject i in trial j for each endpoint during the period k.

Individual-level surrogacy. The estimated variance of the shared frailties u_{ij} is $\hat{\sigma}^2_{\text{indiv}}$ and can be used to estimate the Kendall's $\hat{\tau} = 4\int_0^\infty s\mathcal{L}(s)\mathcal{L}^{(2)}(s)\mathrm{d}s - 1$, where $\mathcal{L}(s)$ and $\mathcal{L}^{(2)}(s)$ are the Laplace transform of the frailty distribution and its second derivative. As an analytic expression of $\mathcal{L}(s)$ is not available for the log-normal frailty distribution, we approximated it using the Laplace method [15], implemented in the `fr.lognormal()` function in the `parfm` package [23, 35].

Trial-level surrogacy. In model (9), the trial-specific treatment effects are again assumed to follow the binormal distribution (6). Thus, the correlation ρ_{trial} between the two treatment effects provides us with the coefficient of determination $R^2_{\text{trial}} = \rho^2_{\text{trial}}$, also referred to simply as R^2.

Reduced Poisson models. The `surrosurv` package can compute four reduced versions of the full model (9) that may turn out to be useful in case of convergence issues with the full model.

- Model **Poisson T** has random trial-treatment interactions α_i and β_i, but does not incorporate individual effects ($u_{ij} \equiv 0$). It assumes common baselines between trials ($\mu_{Si}^{(k)} = \mu_S^{(k)}$, $\mu_{Ti}^{(k)} = \mu_T^{(k)}, \forall i$). This model provides only the trial-level measure of surrogacy R^2_{trial}.
- Model **Poisson I** contains individual random effects u_{ij}, but not the trial-specific treatment effects ($\alpha_i = \alpha$, $\beta_i = \beta, \forall i$) and has common baselines between trials. This model provides only the individual-level measure of surrogacy τ.
- Model **Poisson TI** incorporates both random trial-treatment interactions $(\alpha_i, \beta_i)'$ and individual random effects u_{ij}, but still has common baselines between trials. It provides both individual-level and trial-level measures of surrogacy τ and R^2_{trial}.
- Model **Poisson TIa** extends the model Poisson TI by accounting for trial-specific baseline risks, using shared random effects at the trial level: $\mu_{Si} = \mu_S + m_i, \mu_{Ti} = \mu_T + m_i$, with $m_i \sim \mathcal{N}(0, \sigma_m^2)$.

3. Program description with a data example

We illustrate the use of the functions in the `surrosurv` package on the individual patient data of the advanced GASTRIC meta-analysis [14, 27].

```
> library(surrosurv)
Loading required package: optimx
> packageVersion('surrosurv')
```

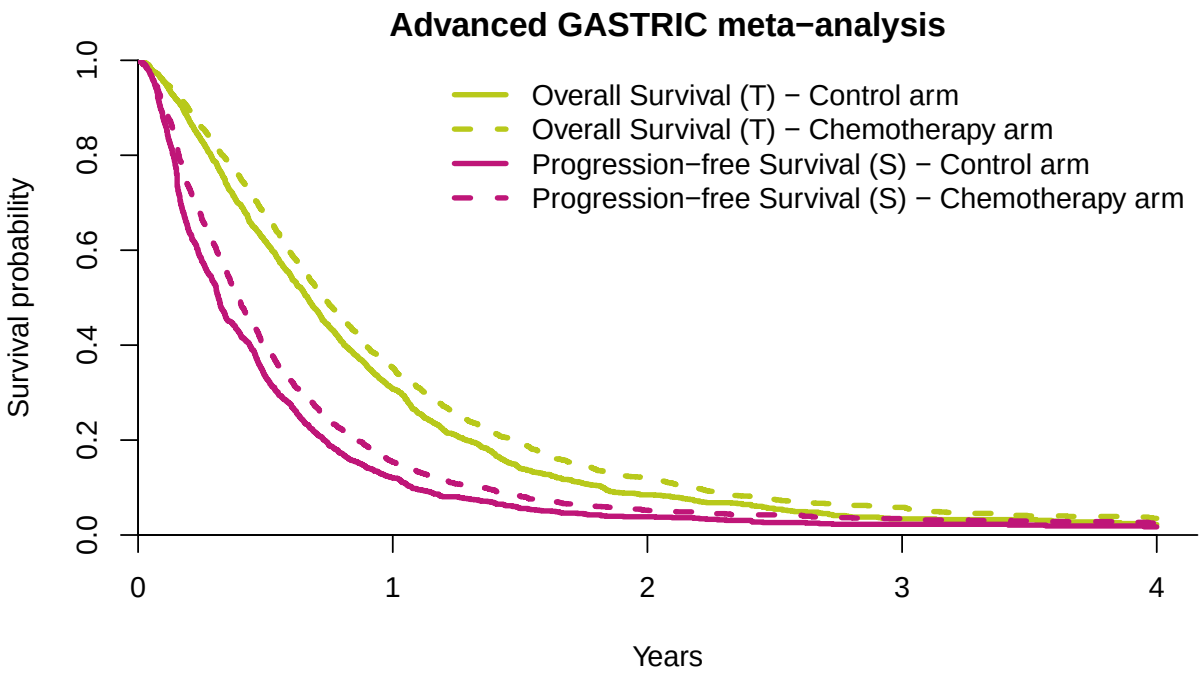

Figure 1: Kaplan–Meier curves for overall survival (T) and progression-free survival (S) in the advanced GASTRIC meta-analysis [14]

```
[1] '1.1.5'
```

The individual data of the 4069 patients, already made public by [6], are also available directly in `R` in the `surrosurv` package:

```
> data('gastadv')
> nrow(gastadv)
[1] 4069
```

The data set contains the following variables:

```
> names(gastadv)
[1] "timeT"   "statusT"   "statusS"   "timeS"   "trialref"   "trt"   "id"
```

where `timeT` and `timeS` are the (possibly censored) times for overall survival (`T`) and for progression-fre survival (`S`) expressed in days, `statusT` and `statusS` are the associated indicators of censoring (`0`) or event (`1`), `trialref` is the trial indicator (i), `trt` is the treatment arm (`-0.5` for control and `0.5` for chemotherapy), and `id` is the patient indicator (j). Figure 1 shows the Kaplan–Meier curves for overall survival, the true endpoint T, and progression-free survival, the candiddate surrogate S.

3.1. Fitting the surrogacy models

The surrogacy models presented in Section 2 can be fitted via the `surrosurv()` function.

The only mandatory argument for the `surrosurv()` function is `data`, which has to be a data.frame with columns

- `trialref`, a factor containing the trial identifier;
- `trt`, the treatment arm, coded as `-0.5` *vs.* `0.5`;
- `id`, a factor containing the patient id;
- `timeT` and `timeS`, two positive-valued numerical variables, containing the observed or censored times of the true endpoint T and of the candidate surrogate S, respectively;
- `statusT` and `statusS`, the censoring/event (`0/1`) indicators of T and S, respectively.

A second argument, `models`, can optionally contain the list of the models to fit (any of `clayton`, `plackett`, `hougaard`, or `poisson`). If not specified, all of them are fitted.

Two further parameters, `intWidth` and `nInts`, specify the width and the number of time intervals for data Poissonization. These parameters are passed to the function `poissonize()`, described in the Appendix (Sec. A). At most one of them can be specified. By default, `nInts = 8` which means that the study period is divided into eight periods, the length of which is fixed so that 1/8th of the observed events falls in each interval.

The optimizer used for optimization of the copula models and the Poisson models can be passed to the `optimx` package [24, 25] via the arguments `cop.OPTIMIZER` and `poi.OPTIMIZER`.

The last parameter, `verbose`, is a logical value stating whether the function should print out the model being fitted (default: `FALSE`).

The surrogacy models for the advanced GASTRIC cancer meta-analysis are obtained as follows:

```
> allSurroRes <-  surrosurv(gastadv, verbose = TRUE)
Estimating model: clayton
Estimating model: plackett
Estimating model: hougaard
Estimating model: poisson
```

Note that the computation time of the surrogacy model estimation can be long. In this example, the computations required 38 mins on a PC with an Intel® quad-core CPU E3-1280 V2 with 3.60 GHz clock speed and 16GB of RAM. The results are an object of class `surrosurv` and the estimated Kendall's τ and R^2 can be easily displayed:

```
>   allSurroRes
                kTau R2
Clayton unadj  0.61 0.45
```

```
Clayton adj     0.61 0.42
Plackett unadj 0.62 0.45
Plackett adj    0.62 0.41
Hougaard unadj 0.32 0.45
Hougaard adj    0.32 0.38
PoissonT        -.-- 1
PoissonI        0.51 -.--
PoissonTI       0.51 0.63
PoissonTIa      0.51 0.83
```

For each copula model, both the results with measurement error adjustment (`adj`) and without adjustment (`unadj`) are shown.

3.1.1. Assessing convergence

The function `convergence()` checks whether convergence criteria are met by each of the fitted models. Three convergence criteria are considered. The first criterion, `maxSgrad`, verifies whether the maximum gradient is small enough. The two other criteria, `minHev` and `minREev`, verify whether the minimum eigenvalue of the Hessian matrix of the fixed parameters (`H`) and of the covariance matrix of the random effects (`RE`) are big enough, in order to assure the positive definitess of the two matrices. Two parameters can be used to tune the thresholds for 'small enough' maximum gradient and for 'big enough' minmum eigen value: `kkttol` (`1e-2` by default), and `kkt2tol` (`1e-8` by default).

```
> convergence(allSurroRes)
               maxSgrad minHev minREev
Clayton unadj     FALSE  FALSE     ---
Clayton adj       FALSE  FALSE    TRUE
Plackett unadj    FALSE  FALSE     ---
Plackett adj      FALSE  FALSE    TRUE
Hougaard unadj    FALSE   TRUE     ---
Hougaard adj      FALSE   TRUE    TRUE
PoissonT           TRUE   TRUE   FALSE
PoissonI           TRUE   TRUE     ---
PoissonTI          TRUE   TRUE    TRUE
PoissonTIa         TRUE   TRUE    TRUE
```

If the values of the minimum gradient and of the maximum eigenvalues are needed, the function `convals()` can be used:

```
> convals(allSurroRes)
               maxSgrad   minHev minREev
Clayton unadj   1.5e+00 -6.1e+00     ---
```

```
Clayton adj      1.5e+00 -6.1e+00 9.8e-03
Plackett unadj   5.9e+02 -5.3e+00     ---
Plackett adj     5.9e+02 -5.3e+00 8.7e-03
Hougaard unadj   1.4e+01  7.7e-01     ---
Hougaard adj     1.4e+01  7.7e-01 7.7e-03
PoissonT         1.3e-05  1.3e+02 6.3e-12
PoissonI         2.0e-05  6.8e+01     ---
PoissonTI        7.1e-06  6.7e+01 2.0e-02
PoissonTIa       5.0e-05  9.4e+07 1.0e-01
```

3.2. Prediction of the treatment effect

When fitting surrogacy models, an estimate of the treatment effects on the two endpoints is computed for each trial. The function `predict()`, applied to an object of class `surrosurv`, returns the predictions of the treatment effects for each trial. The minimal syntax is `predict(allSurroRes)`, but one can be interested in prediction of only one of the fitted models:

```
>   predict(allSurroRes, models = 'PoissonTI')
Treatment effect prediction for surrosurv object
   Poisson TI
                              1     2     3     4     5     6
    Treatment effects on S: -0.52 -0.42 -0.38 -0.08 -0.51 -0.38 ...
    Treatment effects on T: -0.26 -0.08 -0.27  0.41 -0.41 -0.15 ...
```

This function returns an object of class `predictSurrosurv`.

The predicted treatment effects can also be vizualied graphically using the linear regression of the effect on T given the effect on S. The usual surrogacy plot is obtained using the function `plot()` for the classes `surrosurv` and `predictSurrosurv`. For example, the surrogacy plots for the adjusted Clayton copula and the Poisson TI models in the advanced GASTRIC meta-analysis (Fig. 2) can be obtained as follows:

```
>   plot(allSurroRes, c('Clayton adj', 'PoissonTI'))
```

The argument `surro.stats` controls whether the estimated Kendall's τ and R^2 must be displayed on the plots; `pred.ints` controls whether the prediction intervals must be plotted; `show.ste` controls whether the surrogate threshold effect (STE) must be displayed on the plots. The STE is the minimal treatment effect to be observed on the surrogate endpoint S to predict a statistically significant effect on the true endpoint T [1]. The value of the STE estimated by each surrogacy model can be obtained via the function `ste()`, both in terms of regression parameter (`beta`) and in terms of hazard ratio (`HR`):

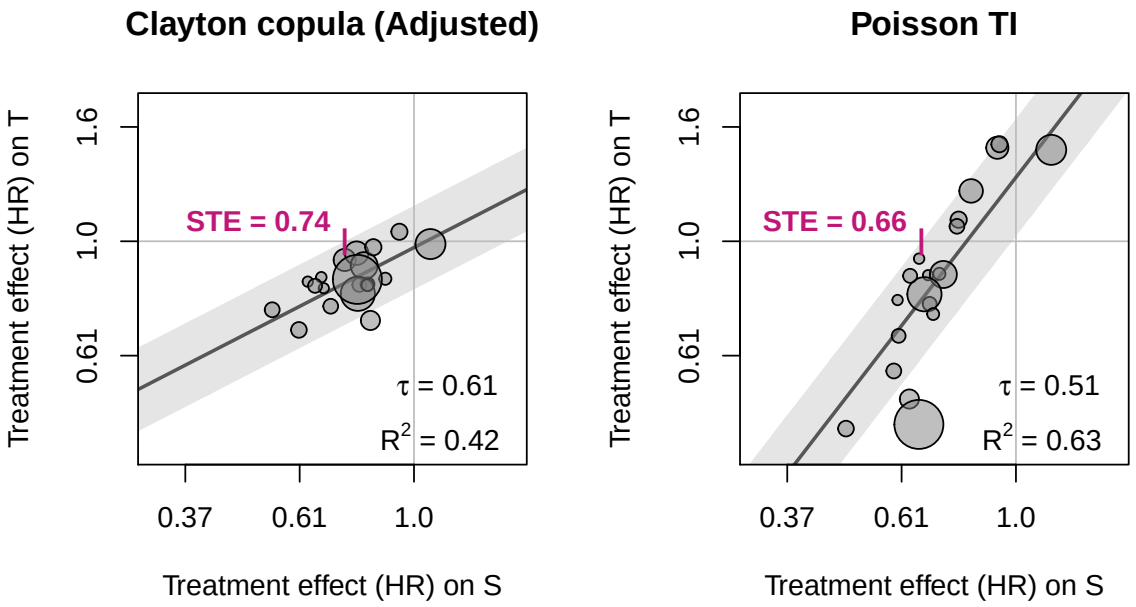

Figure 2: Predictions for the advanced GASTRIC meta-analysis, as computed by the adjusted Clayton copula model, which had poor convergence metrics, and by the Poisson TI model, which was deemed to have converged. HR = hazard ratio.

```
> ste(allSurroRes)
                  beta   HR
Clayton.unadj   -0.61 0.54
Clayton.adj     -0.30 0.74
Plackett.unadj  -0.61 0.54
Plackett.adj    -0.30 0.74
Hougaard.unadj  -0.61 0.54
Hougaard.adj    -0.28 0.76
PoissonT        -0.12 0.88
PoissonTI       -0.41 0.66
PoissonTIa      -1.04 0.36
```

3.2.1. Leave-one-trial-out cross-validation

One technique used to assess the validity of the surrogacy model is to apply the leave-one-out principle to the trials in the meta-analysis. This means that, for each trial, the observed treatment effect on S is compared to its prediction obtained by entering the observed effect on T in the surrogacy model fitted on the other $N-1$ trials. [22, 21, 37]. The function `loovc()` allows performing this evaluation for a given list of models. The cross-validation requires fitting as many models as the number of trials N. As each model is usually very time-consuming to

converge, the function `loovc()` has been implemented to fit the N models by parallel computing. The argument `parallel` is a logical for allowing or not such a parallelization, whereas `nCores` allows specifying the number of cores to use. By default, `parallel = TRUE` and `nCores` is set to the minimum between N and the maximum number of cores on the machine.

```
>   loocvRes <- loocv(gastadv, models = c('Clayton', 'PoissonTI'))
Parallel computing on 8 cores (the total number of cores detected)
```

The results of the crossvalidation can be printed

```
>   loocvRes
   Clayton copula (Unadjusted)
         1     2     3     4     5     6
obsBeta -0.31 -0.21 -0.09 -0.02 -0.22 -0.34 ...
lwr     -0.76 -0.65 -0.42 -0.51 -0.48 -0.62 ...
upr     -0.05  0.02  0.28  0.17  0.21  0.09 ...

   Clayton copula (Adjusted)
         1      2      3      4      5      6
obsBeta -0.309 -0.212 -0.095 -0.023 -0.222 -0.342 ...
lwr     -0.571 -0.491 -0.277 -0.358 -0.332 -0.448 ...
upr     -0.213 -0.130  0.105 -0.001  0.042 -0.078 ...

   Poisson TI
         1     2     3     4     5     6
obsBeta -0.31 -0.21 -0.09 -0.02 -0.22 -0.34 ...
lwr     -0.87 -0.67 -0.42 -0.59 -0.29 -0.54 ...
upr     -0.45 -0.21  0.57  0.23  0.23 -0.22 ...
```

and plotted (Fig. 3) by showing, for each trial, the comparison between the observed treatment effect on T, and its prediction interval, based on the observed treatment effect on S for the same trial and the surrogacy model fitted on the other $N - 1$ trials:

```
>   plot(loocvRes)
```

3.3. Utilities for data simulation

Few publications present simulation approaches adapted to discuss statistical methods for evaluating failure time surrogate endpoints [2, 38, 30, 31, 32]. To our knowledge, the data generation methods used to date are based either on the use of a Clayton copula or on a mixture of half-normal and exponential random variables. Thanks to the `surrosurv` package, data can be generated using these two methods, in addition to an approach based on mixed proportional

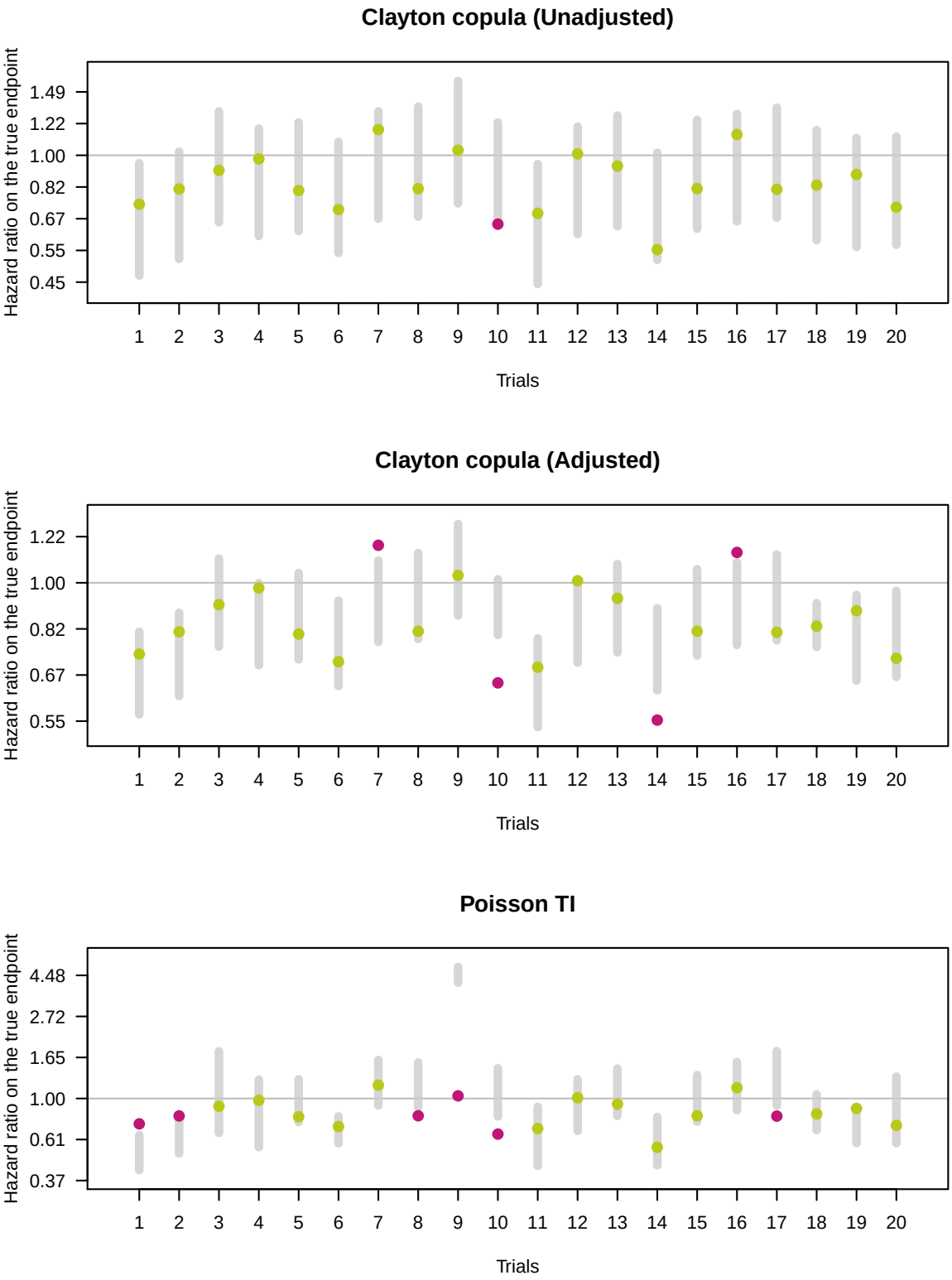

Figure 3: Leave-one-trial-out cross-validation results for the advanced GASTRIC meta-analysis. Vertical lines are the 95% prediction intervals (PI) of the treatment effect on overall survival (OS). Dots are the observed treatment effects on OS (green = within the PI, magenta = out of the PI).

hazard models that we employed recently [36]. These three data generation algorithms are detailed here below.

3.3.1. Data generation based on a Clayton copula

The data geration method used in [2] and in [31, 32] reflects the data generating process underlying the two-step copula model (Sec. 2.1).

We implemented this approach for the Clayton family (Eq. (2)), which is available using the function `simData.cc()`. This function generates data as follows:

- trial-specific random effects are generated from

$$\begin{pmatrix} m_{S_i} \\ m_{T_i} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_S^2 & \sigma_S \sigma_T \rho_m \\ \sigma_S \sigma_T \rho_m & \sigma_T^2 \end{pmatrix} \right)$$

- trial-specific treatment effects are generated from

$$\begin{pmatrix} \alpha_i \\ \beta_i \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \begin{pmatrix} d_a^2 & d_a d_b \rho_{\text{trial}} \\ d_a d_b \rho_{\text{trial}} & d_b^2 \end{pmatrix} \right)$$

- exponentially distributed individual times are simulated for S, conditionally on the random effects generated before.

$$S_{ij} = -\log(U_{Sij})/\lambda_{Sij}, \quad \text{with } \lambda_{Sij} = \exp(\mu_S + m_{S_i} + \alpha_i Z_{ij}) \text{ and } U_{Sij} \sim U(0,1)$$

- exponentially distributed individual times are simulated for $T \mid S$, conditionally on the random effects generated before *and on the value of* S

$$\begin{aligned} T_{ij} \mid S_{ij} = -\log(U'_{Tij})/\lambda_{Tij}, \quad \text{with } & \lambda_{Tij} = \exp(\mu_T + m_{T_i} + \beta_i Z_{ij}), \\ & U'_{Tij} = \left[\left(U_{Tij}^{-\theta/(1+\theta)} - 1 \right) U_{Sij}^{-\theta} + 1 \right]^{-1/\theta}, \text{ and} \\ & U_{Tij} \sim U(0,1). \end{aligned}$$

The details of the arguments of the `simData.cc()` function can be obtained using `help(simData.cc)`.

3.3.2. Data generation based on a mixture of half-normal and exponential random variables

The data geration method used in [38] and in [30] is based on the results by Cowles [8], which showed that a Weibull distribution can be expressed as a scaled mixture of half-normal distribution and an exponential distribution with unit rate parameter.

This approach is implemented in the function `simData.mx()` and generates data as follows:

- trial-specific random effects are generated from

$$\begin{pmatrix} m_{S_i} \\ m_{T_i} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_S^2 & \sigma_S \sigma_T \rho_m \\ \sigma_S \sigma_T \rho_m & \sigma_T^2 \end{pmatrix} \right)$$

- trial-specific treatment effects are generated from

$$\begin{pmatrix} \alpha_i \\ \beta_i \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \begin{pmatrix} d_a^2 & d_a d_b \rho_{\text{trial}} \\ d_a d_b \rho_{\text{trial}} & d_b^2 \end{pmatrix} \right)$$

- individual half-normal random variables Y_{ij}^* are generated from the distribution

$$f(y^*) = \frac{2}{\sqrt{2\pi}} \exp\left(-\frac{y^{*2}}{2}\right), \qquad y^* \in \mathbb{R}_+$$

- unit rate parameter exponential random variables Λ_{Sij} and Λ_{Tij} are generated from $-\log(U_{Sij})_{Sij}$ and $-\log(U_{Tij})$, with $U_{Sij} \sim U(0,1)$ and $U_{Tij} \sim U(0,1)$

- exponentially distributed individual times are simulated for S and T from

$$S_{ij} = \left(Y_{ij}^* \sqrt{2\Lambda_{Sij}}\right) \exp(\mu_S + m_{S_i} + \alpha_i Z_{ij}),$$
$$T_{ij} = \left(Y_{ij}^* \sqrt{2\Lambda_{Tij}}\right) \exp(\mu_S + m_{T_i} + \alpha_i Z_{ij}).$$

The details of the arguments can be obtained using `help(simData.mx)`.

3.3.3. Data generation based on mixed proportional hazard models

Recently we also generated data using individual random effects to control individual-level surrogacy [36]. This approach is implemented in the function `simData.re()` and generates data as follows:

- trial-specific random effects and trial-specific treatment effects were generated as in the Clayton copula case

- individual random effects were generated from $u_{ij} \sim \mathcal{N}(0, \sigma^2)$, with σ^2 depending on the scenario (according to the Kendall's τ)

- exponentially distributed individual times were simulated for S and T, conditionally on the random effects generated before. We used the inverse transform method, which consists in transforming a uniform random variable by means of the inverse of the probability distribution function of the random variable to be generated [see for instance 33, § 2.1.2]

$$S_{ij} = -\log(U_{Sij})/\lambda_{Sij}, \quad \text{with } \lambda_{Sij} = \exp(\mu_S + m_{S_i} + \alpha_i Z_{ij} + u_{ij}) \text{ and } U_{Sij} \sim U(0,1),$$
$$T_{ij} = -\log(U_{Tij})/\lambda_{Tij}, \quad \text{with } \lambda_{Tij} = \exp(\mu_T + m_{T_i} + \beta_i Z_{ij} + u_{ij}) \text{ and } U_{Tij} \sim U(0,1).$$

The details of the arguments can be obtained using `help(simData.re)`.

4. Mode of availability of the `surrosurv` package

The `surrosurv` package is an open-source project. Stable versions are released via the Comprehensive `R` Archive Network (CRAN, `https://cran.r-project.org/package=surrosurv`). Source code is available on the R-forge platform (`https://r-forge.r-project.org/projects/surrosurv/`).

Acknowledgments

The present work has been supported by the Institut National du Cancer (INCa), Grant SHS 2014-141, and by the Ligue Nationale Contre le Cancer. The study sponsors had no involvement in either the study design; the collection, analysis and interpretation of data; the writing of the manuscript; nor in the decision to submit the manuscript for publication. The authors thank the GASTRIC (Global Advanced/Adjuvant Stomach Tumor Research International Collaboration) Group for permission to use their data. The investigators who contributed to GASTRIC are listed in references [26, 27, 13, 14]. The GASTRIC Group data are available within the `surrosurv` package for research purposes, under the conditions that (1) the research be scientifically appropriate, (2) the confidentiality of individual patient data be protected, (3) the results of the analyses be shared with the GASTRIC Group prior to public communication, (4) the source of data be fully acknowledged as above, and (5) resulting data and results be further shared with the research community.

References

[1] T. Burzykowski and M. Buyse. Surrogate threshold effect: An alternative measure for meta-analytic surrogate endpoint validation. *Pharmaceutical Statistics*, 5(3):173–186, July 2006. doi:10.1002/pst.207.

[2] T. Burzykowski and J. Cortiñas Abrahantes. Validation in the case of two failure-time endpoints. In T. Burzykowski, G. Molenberghs, and M. Buyse, editors, *The Evaluation of Surrogate Endpoints*, pages 163–194. Springer, New York, NY, 2005.

[3] T. Burzykowski, G. Molenberghs, M. Buyse, H. Geys, and D. Renard. Validation of surrogate end points in multiple randomized clinical trials with failure time end points. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 50(4):405–422, 2001. doi:10.1111/1467-9876.00244.

[4] T. Burzykowski, G. Molenberghs, and M. Buyse. *The Evaluation of Surrogate Endpoints*. Springer Science & Business Media, New York, NY, 2006.

[5] M. Buyse, G. Molenberghs, T. Burzykowski, D. Renard, and H. Geys. The validation of surrogate endpoints in meta-analyses of randomized experiments. *Biostatistics*, 1(1):49–67, 2000. doi:10.1093/biostatistics/1.1.49.

[6] M. Buyse, G. Molenberghs, X. Paoletti, K. Oba, A. Alonso, W. Van der Elst, and T. Burzykowski. Statistical evaluation of surrogate endpoints with examples from cancer clinical trials. *Biometrical Journal*, 58(1):104–132, 2016. doi:10.1002/bimj.201400049.

[7] D. Clayton. A model for association in bivariate life tables and its application in epidemiological studies of familial tendency in chronic disease incidence. *Biometrika*, 65:141–151, 1978. doi:10.1093/biomet/65.1.141.

[8] M. Cowles. Evaluating surrogate endpoints for clinical trials: A bayesian approach. Technical report, University of Iowa, 2004. URL `https://stat.uiowa.edu/sites/stat.uiowa.edu/files/techrep/tr334.pdf`.

[9] M. J. Crowther, R. D. Riley, J. A. Staessen, J. Wang, F. Gueyffier, and P. C. Lambert. Individual patient data meta-analysis of survival data using poisson regression models. *BMC Medical Research Methodology*, 12(1):34, 2012. doi:10.1186/1471-2288-12-34.

[10] L. Duchateau and P. Janssen. *The Frailty Model*. Springer Verlag, New York, NY, 2008. ISBN 978-0-387-72834-6. doi:10.1007/978-0-387-72835-3.

[11] A. Gasparrini. `mvmeta`: Multivariate and univariate meta-analysis and meta-regression. `R` package version 0.4-7, 2015. URL `https://CRAN.R-project.org/package=mvmeta`.

[12] A. Gasparrini, B. Armstrong, and M. Kenward. Multivariate meta-analysis for non-linear and other multi-parameter associations. *Statistics in Medicine*, 31:3821–3839, 2012. doi:10.1002/sim.5471.

[13] GASTRIC group. Benefit of adjuvant chemotherapy for resectable gastric cancer: A meta-analysis. *Journal of the American Medical Association*, 303(17):1729–1737, 2010. doi:10.1001/jama.2010.534.

[14] GASTRIC group. Role of chemotherapy for advanced/recurrent gastric cancer: An individual-patient-data meta-analysis. *European Journal of Cancer*, 49(7):1565–1577, 2013. doi:10.1016/j.ejca.2012.12.016.

[15] C. Goutis and G. Casella. Explaining the saddlepoint approximation. *The American Statistician*, 53(3):216–224, 1999. doi:10.1080/00031305.1999.10474463.

[16] M. Hofert, I. Kojadinovic, M. Maechler, and J. Yan. `copula`: Multivariate dependence with copulas. `R` package version 0.999-15, 2016. URL `https://CRAN.R-project.org/package=copula`.

[17] P. Hougaard. A class of multivariate failure time distributions. *Biometrika*, 73:671–678, 1986. doi:10.1093/biomet/73.3.671.

[18] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938. URL `http://www.jstor.org/stable/2332226`.

[19] S. Kovalchik. Reply to exponential proportional hazard model. `http://r.789695.n4.nabble.com/exponential-proportional-hazard-model-td805536.html`, 2013. Accessed: 2017-02-10.

[20] N. Laird and D. Olivier. Covariance analysis of censored survival data using log-linear analysis techniques. *Journal of the American Statistical Association*, 76(374):231–40, 1981. doi:10.1080/01621459.1981.10477634.

[21] A. Mauguen, J.-P. Pignon, S. Burdett, C. Domerg, D. Fisher, R. Paulus, S. J. Mandrekar, C. P. Belani, F. A. Shepherd, T. Eisen, H. Pang, L. Collette, W. T. Sause, S. E. Dahlberg, J. Crawford, M. O'Brien, S. E. Schild, M. Parmar, J. F. Tierney, C. Le Pechoux, and S. Michiels. Surrogate endpoints for overall survival in chemotherapy and radiotherapy trials in operable and locally advanced lung cancer: A re-analysis of meta-analyses of individual patients' data. *The Lancet Oncology*, 14(7):619–26, 2013. doi:10.1016/S1470-2045(13)70158-X.

[22] S. Michiels, A. Le Maître, M. Buyse, T. Burzykowski, E. Maillard, J. Bogaerts, J. B. Vermorken, W. Budach, T. F. Pajak, K. K. Ang, J. Bourhis, and J.-P. Pignon. Surrogate endpoints for overall survival in locally advanced head and neck cancer: meta-analyses of individual patient data. *The Lancet Oncology*, 10(4):341–50, 2009. doi:10.1016/S1470-2045(09)70023-3.

[23] M. Munda, F. Rotolo, and C. Legrand. `parfm`: Parametric frailty models in `R`. *Journal of Statistical Software*, 51(1):1–20, 2012. doi:10.18637/jss.v051.i11.

[24] J. C. Nash. On best practice optimization methods in `R`. *Journal of Statistical Software*, 60(2):1–14, 2012. doi:10.18637/jss.v043.i09.

[25] J. C. Nash, R. Varadhan, and G. Grothendieck. `opitmx`: A replacement and extension of the `optim()` function. `R` package version 2013.8.7, 2013. URL `https://CRAN.R-project.org/package=optimx`.

[26] K. Oba, X. Paoletti, S. Alberts, Y.-J. Bang, J. Benedetti, H. Bleiberg, P. Catalano, F. Lordick, S. Michiels, S. Morita, Y. Ohashi, J.-P. Pignon, P. Rougier, M. Sasako, J. Sakamoto, D. Sargent, K. Shitara, E. Van Cutsem, M. Buyse, and T. Burzykowski. Disease-free survival as a surrogate for overall survival in adjuvant trials of gastric cancer: A meta-analysis. *Journal of the National Cancer Institute*, 105(21):1600–7, 2013. doi:10.1093/jnci/djt270.

[27] X. Paoletti, K. Oba, Y.-J. Bang, H. Bleiberg, N. Boku, O. Bouché, P. Catalano, N. Fuse, S. Michiels, M. Moehler, S. Morita, Y. Ohashi, A. Ohtsu, A. Roth, P. Rougier, J. Sakamoto, D. Sargent, M. Sasako, K. Shitara, P. Thuss-Patience, E. Van Cutsem, T. Burzykowski, and M. Buyse. Progression-Free Survival as a Surrogate for Overall Survival in Advanced/Recurrent Gastric Cancer Trials: A Meta-Analysis. *Journal of the National Cancer Institute*, 105(21):1667–1670, 2013. doi:10.1093/jnci/djt269.

[28] R. L. Plackett. A class of bivariate distributions. *Journal of the American Statistical Association*, 60:516–522, 1965. doi:10.1080/01621459.1965.10480807.

[29] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016. URL `https://www.R-project.org/`.

[30] L. A. Renfro, Q. Shi, D. J. Sargent, and B. P. Carlin. Bayesian adjusted R^2 for the meta-analytic evaluation of surrogate time-to-event endpoints in clinical trials. *Statistics in Medicine*, 31(8):743–761, 2012. doi:10.1002/sim.4416.

[31] L. A. Renfro, Q. Shi, Y. Xue, J. Li, H. Shang, and D. J. Sargent. Center-within-trial versus trial-level evaluation of surrogate endpoints. *Computational Statistics & Data Analysis*, 78: 1–20, 2014. doi:10.1016/j.csda.2014.03.011.

[32] L. A. Renfro, H. Shang, and D. J. Sargent. Impact of copula directional specification on multi-trial evaluation of surrogate endpoints. *Journal of Biopharmaceutical Statistics*, 25: 857–877, 2015. doi:10.1080/10543406.2014.920870.

[33] C. Robert and G. Casella. *Introducing Monte Carlo Methods with R*. Springer Science & Business Media, 2009. doi:10.1007/978-1-4419-1576-4.

[34] F. Rotolo. `surrosurv`: Evaluation of failure time surrogate endpoints in individual patient data meta-analyses. `R` package version 1.1.5, 2016. URL `https://CRAN.R-project.org/package=surrosurv`.

[35] F. Rotolo and M. Munda. `parfm`: Parametric frailty models. `R` package version 2.5.10, 2015. URL `https://CRAN.R-project.org/package=parfm`.

[36] F. Rotolo, X. Paoletti, M. Buyse, T. Burzykowski, and S. Michiels. Evaluation of failure time surrogate endpoints in individual patient data meta-analyses of randomized clinical trials: a poisson approach. *Under Review*, 2017.

[37] F. Rotolo, J.-P. Pignon, J. Bourhis, S. Marguet, J. Leclercq, W. T. Ng, J. Ma, A. Chan, P.-Y. Huang6, G. Zhu, D. Chua, Y. Chen, H. Q. Mai, D. L. W. Kwong, Y. L. Soong, J. Moon, Y. Tung, K. H. Chi, G. Fountzilas, L. Zhang, E. Pun Hui, A. W. M. Lee, P. Blanchard, and S. Michiels. Surrogate endpoints for overall survival in loco-regionally advanced nasopharyngeal carcinoma: an individual patient data meta-analysis. *Journal of the National Cancer Institute*, 109(4), 2017. doi:10.1093/jnci/djw239.

[38] Q. Shi, L. A. Renfro, B. M. Bot, T. Burzykowski, M. Buyse, and D. J. Sargent. Comparative assessment of trial-level surrogacy measures for candidate time-to-event surrogate endpoints in clinical trials. *Computational Statistics & Data Analysis*, 55(9):2748–2757, 2011. doi:10.1016/j.csda.2011.03.014.

[39] H. C. van Houwelingen, L. Arends, and T. Stijnen. Advanced methods in meta-analysis: Multivariate approach and meta-regression. *Statistics in Medicine*, 21:589–624, 2002. doi:10.1002/sim.1040.

[40] J. Whitehead. Fitting Cox's regression model to survival data using `GLIM`. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, pages 268–275, 1980. doi:10.2307/2346901.

[41] J. Yan. Enjoy the joy of copulas: With a package `copula`. *Journal of Statistical Software*, 21(1):1–21, 2007. doi:10.18637/jss.v021.i04.

A. Data poissonization

Fitting auxiliary Poisson models for estimating the parameters of a proportional hazard model [40, 9] needs that data are rearranged in order to provide, for each time period, the number of events and the total time passed at risk. The function `poissonize()` in the `surrosurv` package allows to perform the necesasry data manipulaton. The core of the function has been derived from the original code publicly shared by [19].

The main argument of the `poissonize()` function is `data`, a data frame with columns: `id`, the patient identifyier; `time`, the event/censoring time; `status`, the event (`1`) or censoring (`0`) indicator; `...`, other factors such like the covariables needed in the regression model.

The breakpoints between time intervals can be entered in the second argument, `all.breaks`. Otherwise, if `all.breaks` is not specified, one can specify either the width of the time intervals `interval.width`, or their number `nInts` (used only if also `is.null(interval.width)`).

Any other variables to be kept in the poissonized data frame can be entered in `factors`. The last argument (`compress`) is a logical value indicating whether the record with the same factor profile should be summarized into one record, i. e. whether the data should be expressed in a short form.

In the advanced GASTRIC cancer example, we first change the column names in order to match the ones needed by `poissonize()`:

```
>   gastadv.poi <- gastadv
>   gastadv.poi$time <- gastadv.poi$timeT / 365.25
>   gastadv.poi$status <- gastadv.poi$statusT
```

We fit the proportional hazard model, to which we will compare the results of the auxiliary Poisson model

```
> fitcox <- coxph(Surv(time, status) ~ trt, data = gastadv.poi)
> cox.base <- basehaz(fitcox, centered = FALSE)
```

and we plot the estimated survival curves.

```
> plot(stepfun(cox.base$time[-nrow(cox.base)],
+                exp(-cox.base$hazard)),
+      ylim = 0:1, yaxs = 'i',
+      xlim = c(0, 5), xaxs = 'i',
+      col = 1, lwd = 2, bty = 'l',
+      do.points = FALSE, verticals = FALSE,
+      main = 'Overall Survival\nAdvanced GASTRIC meta-analysis',
+      xlab = 'Years', ylab = 'Survival probability')
> lines(stepfun(cox.base$time[-nrow(cox.base)],
+                exp(-cox.base$hazard * exp(coef(fitcox)['trt']))),
+      col = 2, pch = '', lwd = 2)
```

We 'possonize' the data over 10 intervals (the default) and we fit the auxiliary Poisson model.

```
> gastadv.poi <- poissonize(gastadv.poi, nInts = 10, factors = 'trt')
> gastadv.poi
             interval  trt   m  Rt    N
1                   0 -0.5 181 292 1668
2    0.1832128678987 -0.5 180 173 1475
3   0.30921697467488 -0.5 192 149 1288
4   0.435221081451061 -0.5 159 132 1088
5   0.567018480492813 -0.5 154 114  912
6   0.703885010266941 -0.5 156 108  751
7   0.867545516769336 -0.5 157 103  584
8    1.07320739219713 -0.5 143 101  414
9    1.39328678986995 -0.5 117  97  239
10   2.07255030800821 -0.5  60  87   94
11                  0  0.5 216 421 2401
12   0.1832128678987  0.5 221 258 2167
13  0.30921697467488  0.5 213 229 1935
14 0.435221081451061  0.5 247 207 1706
15 0.567018480492813  0.5 237 181 1446
16 0.703885010266941  0.5 225 176 1203
17 0.867545516769336  0.5 228 171  965
18   1.07320739219713  0.5 221 183  715
19   1.39328678986995  0.5 211 205  460
20   2.07255030800821  0.5 117 171  204
```

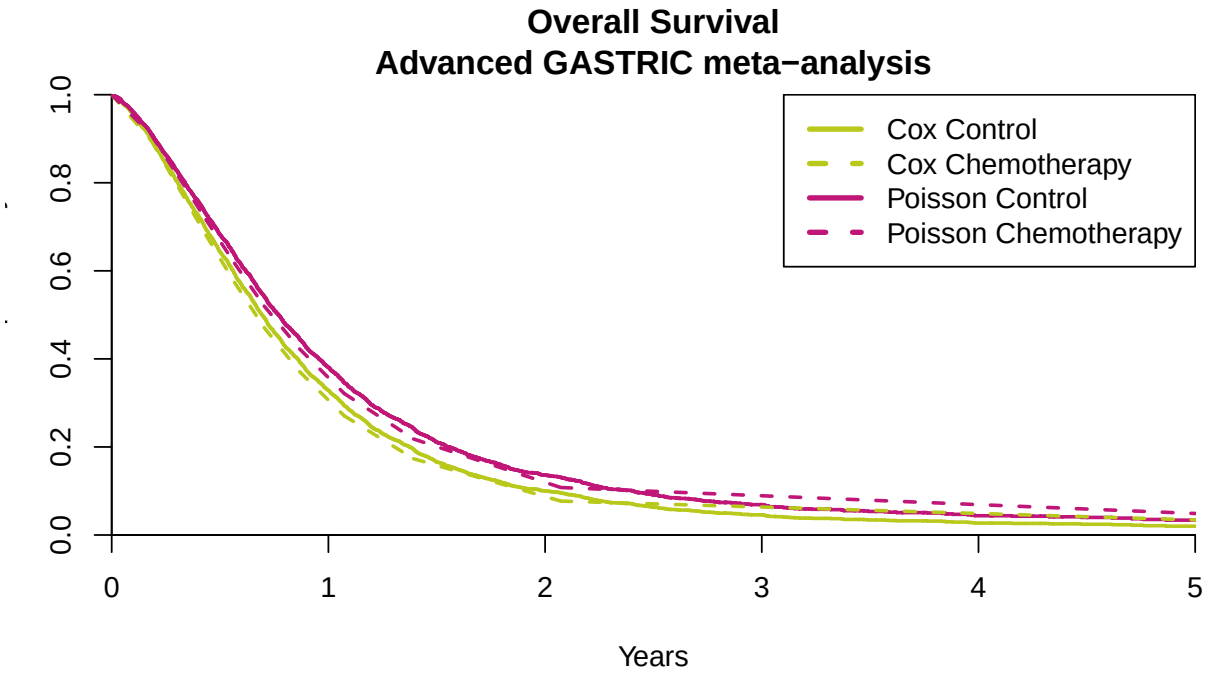

Figure 4: Overall survival curves in the advanced GASTRIC meta-analysis. Comparison between the survival probability obtained using the Breslow estimator in the Cox model (solid lines) and those obtained using the auxiliary Poisson model (dashed lines).

```
> fitpoi <- glm(m ~ -1 + interval + trt + offset(log(Rt)),
+               data = gastadv.poi, fam = 'poisson')
```

The function `plotsson()` can be used to draw the survival curves (or the instantaneous hazard) estimated by the auxiliary Poisson model:

```
> plotsson(fitpoi, 'Surv', add = TRUE, lty = 2, by = 'trt', lwd = 2)
> legend('topright', col = rep(1:2, each = 2), lty = 1:2, lwd = 3,
+        legend = t(outer(c('Cox', 'Poisson'),
+                         c('Control', 'Chemotherapy'), paste)))
```

The option `add = TRUE` is used to add the curves to the plot from the Cox estimates drawn previously.

The treatment effect estimated by the Cox model is -0.14 (SE $= 0.03$), and it is -0.14 (SE $= 0.03$) when using the auxiliary Poisson model.