Unified ROC Curve Estimator for Diagnosis and Prognosis Studies: The sMSROC Package

by Susana Díaz-Coto, Pablo Martínez-Camblor, and Norberto Corral-Blanco

Abstract The binary classification problem is a hot topic in Statistics. Its close relationship with the diagnosis and the prognosis of diseases makes it crucial in biomedical research. In this context, it is important to identify biomarkers that may help to classify individuals into different classes, for example, diseased vs. not diseased. The Receiver Operating-Characteristic (ROC) curve is a graphical tool commonly used to assess the accuracy of such classification. Given the diverse nature of diagnosis and prognosis problems, the ROC curve estimation has been tackled from separate perspectives in each setting. The Two-stages Mixed-Subjects (sMS) ROC curve estimator fits both scenarios. Besides, it can handle data with missing or incomplete outcome values. This paper introduces the R package sMSROC which implements the sMS ROC estimator, and includes tools that may support researchers in their decision making. Its practical application is illustrated on three real-world datasets.

1 Introduction

The binary classification problem is crucial in biomedical environments. Researchers and physicians face the task of classifying patients (as e.g. diseased vs. disease-free, at risk vs. risk-free, etc.) daily. Frequently, biological measures (biomarkers) are used as objective medical signs that may indicate, for example, the presence/progress of an event of interest or the response to specific treatments. Diagnostic biomarkers are normally employed to detect the presence of a disease. For example, the *circulating cardiac troponin I* aids in noninvasive detection of myocardial injury in cardiovascular diseases (Ni and Wehrens, 2018). The *blood glucose* and the *hemoglobin A1c* are recognized diagnostic biomarkers of type 2 diabetes mellitus (Long et al., 2020) and the *sweat chloride* is often used to confirm the cystic fibrosis (Farrell et al., 2008).

The Receiver Operating-Characteristic (ROC) curve is a popular graphical tool for assessing the ability of biomarkers to discriminate between positive and negative individuals (with and without the event of interest, respectively) (Zhou et al., 2002; Pepe, 2003). For each possible cut-off point, it plots the pairs formed by the complement of *specificity* against the *sensitivity*. The sensitivity, Se, and the specificity, Sp, are the proportions of positive and negative individuals, respectively, who have been correctly classified. The closer the ROC curve to the upper left corner, the more accurate the biomarker.

Mathematically, for each $u \in [0,1]$, the ROC curve has the expression:

$$\mathcal{R}(u) = Se\left(Sp^{-1}(1-u)\right), \tag{1}$$

where $Sp^{-1}(\cdot) = \inf\{x : Sp(x) \geq \cdot\}$. Conventionally, it is assumed that individuals with larger biomarker values are more likely positive. Therefore, the classification sets defining an individual as positive are those in the way $[c, \infty)$, with $c = Sp^{-1}(1-u) \in \mathbb{R}$. Besides, the related area under the ROC curve, (AUC) $\left(= \int_0^1 \mathcal{R}(u)du\right)$, is commonly used as a summary index of the global classification accuracy (Hanley and McNeil, 1982).

The estimation of ROC curves has been addressed from different perspectives (see, for instance, Gonçalves et al. (2014) and references therein). Estimation procedures generally assume that data come from case-control designs, where the actual status of all individuals as positive or negative is known in advance. Further, they do not handle the issue of potential missing values in the outcome. Notice that such issues arise, for instance, in cohort designs where the outcome is defined ad hoc through a subsidiary variable. The missing values for that variable lead to missing values in the outcome.

Most of the statistical software (SPSS, SAS, STATA, etc.) offer routines for computing different ROC curve estimators and related elements. There are also packages in R (www.r-project.org) dealing with the ROC curve topic. We highlight the **pROC** package (Robin et al., 2011), which provides functions for visualizing, smoothing and comparing ROC curves; the package **nsROC** (Pérez-Fernández, 2017), where some of the non-standard tools for the ROC curve analyses described in Pérez-Fernández et al. (2018) are implemented; the package **ROCR** (Sing et al., 2005), which supplies user-friendly tools for creating graphics for visualizing classification performance (the ROC curve is a particular case); the **plotROC** package (Sachs, 2017), offering interactive ROC curve plots suitable for use on the web, and

finally the **ROCnReg** package (Rodríguez-Álvarez and Inácio, 2021), which, among other extensive functionality, implements Bayesian methods for the estimation of ROC curves.

Prognostic biomarkers are used to identify individuals who are likely to experience a future clinical event (death, the onset/recurrence of a disease or the development of a new medical condition). For example, the breast cancer genes 1 and 2 mutations are often employed to assess the likelihood of a second breast cancer (Basu et al., 2015). Similarly, the C-reactive protein level is a prognostic biomarker used to identify individuals with unstable angina at risk of developing other adverse events (Ferreirós et al., 1999), and the Gleason score helps to assess the likelihood of prostate cancer progression (Epstein et al., 2016). These biomarkers are measured at baseline and individuals are then followed over time to observe whether or not the event of interest occurs. A *time-to-event* variable is involved in this process. Different definitions of positive and negative outcomes have been proposed, which has given rise to extensions of the sensitivity and specificity measures and, of course, to the corresponding time-dependent ROC curves (Etzioni et al., 1999) .

The Cumulative/Dynamic (C/D) ROC curve (Heagerty et al., 2000) is perhaps the most natural extension of the ROC curve for time-dependent outcomes. Once set to a specific point of time t, the time-to-event variable is reduced to a dichotomous variable at that time. Then, the sensitivity and the specificity can be extended to the so-called *cumulative sensitivity* and *dynamic specificity* whose expressions are

$$Se_t^{\mathbb{C}}(c) = \mathcal{P}(X > c \mid T \leq t),$$
$$Sp_t^{\mathbb{D}}(c) = \mathcal{P}(X \leq c \mid T > t),$$

where $c \in \mathbb{R}$ is the cut-off point, and X and T are the random variables modeling the biomarker and the time-to-event variables, respectively. The C/D ROC curve is the plot of the pairs formed by the complement to the dynamic specificity and the cumulative sensitivity, for all possible cut-off points. Alternatively, it is given by

$$\mathcal{R}_t^{\mathbb{C}/\mathbb{D}}(u) = Se_t^{\mathbb{C}}\left([Sp_t^{\mathbb{D}}]^{-1}(1-u)\right), \quad u \in [0,1].$$

The area under the C/D ROC curve is used as well as a summarize index of the prognostic accuracy of a biomarker and has the expression

$$AUC_t^{\mathbb{C}/\mathbb{D}} = \int_0^1 \mathcal{R}_t^{\mathbb{C}/\mathbb{D}}(u)du.$$

The main challenge when estimating the C/D ROC curve is the potential lack of complete information for some individuals (caused by censoring). It arises because of loss of follow-up, either due to dropouts or because the study ended before the event of interest had the chance to occur in the individual (right censoring). It may also come up when individuals are not constantly monitored and the only available information is that the event of interest occurred between two observed timepoints (interval censorship). The simplest C/D ROC curve estimator removes from the sample the censored observations and approximates the cumulative sensitivity and the dynamic specificity through their empirical estimators (naive method). Other procedures integrate, in some way, the information from the censored observations. Kamarudin et al. (2017) provides an illustrative revision of the available R packages implementing some of these methods, all of them addressing the right censorship problem. For example, the **survivalROC** package (Heagerty and Saha-Chaudhuri, 2022) computes the C/D ROC curve through the two procedures proposed by Heagerty et al. (2000). The **survAUC** package (Potapov et al., 2023) collects several routines for computing the $AUC_t^{\mathbb{C}/\mathbb{D}}$, at different times, estimated by the Inverse Probability Censoring Weighting (IPCW) method (Uno et al., 2007; Hung and Chiang, 2010) and by the Chambles and Diao (2006) approach. The **timeROC** package (Blanche et al., 2013b) implements the Conditional Inverse Probability Censoring Weighting (CIPCW) procedure (Blanche et al., 2013a). In addition to confidence intervals for the $AUC_t^{\mathbb{C}/\mathbb{D}}$, the package performs tests for comparing two areas under the curve corresponding to different prognostic biomarkers. We add to this list the already mentioned **nsROC** package, which allows to compute the estimation procedures proposed in Martínez-Camblor et al. (2016) and Li et al. (2018). The latter method is also available in the **tdROC** package (Li and Wu, 2016). Finally, the **smoothROCtime** (Díaz-Coto et al., 2020b) and **cenROC** (Beyene and El Ghouch, 2023) packages implement the smooth C/D ROC curve estimators suggested by Martínez-Camblor and Pardo-Fernández (2018) and Beyene and El Grouch (2020), respectively. We only found two packages implementing the C/D ROC curve estimation under interval censorship. The **intcensROC** package (Lin et al., 2021) computes the estimator for the C/D ROC curve and $AUC_t^{\mathbb{C}/\mathbb{D}}$ proposed in Wu et al. (2020), while the **cenROC** implements the method proposed in Beyene and El Ghouch (2022). Not a package but an R function is provided in Díaz-Coto et al. (2020a), computing the C/D ROC curve estimator proposed in that paper.

We present here the package **sMSROC**, which implements the so-called Two-stage Mixed-Subjects

(sMS) estimator. The sMS estimator uses, in a first stage, a probabilistic model for linking the biomarker with the outcome, and then, in a second stage and for each potential threshold, it computes both the sensitivity and the specificity values, which can be used to draw the ROC curve. This approach can be used to answer both diagnostic and prognostic questions, and, by imposing additional constraints on the missing-value mechanism, is able to handle missing data in the outcome variable. The probabilistic model is used for allocating subjects into the positive and the negative groups with certain probabilities. In this sense, subjects are simultaneously classified as positive and negative, that is, they are mixed. Interested readers are referred to Díaz-Coto et al. (2021) for a more in depth explanation of the theoretical properties of the sMS estimator. The presented **sMSROC** package offers a set of exploratory tools which help to choose the most suitable probabilistic model (logistic regression, proportional hazard Cox regression, etc.). These (more standard) estimation proposals are already implemented in the package, which also allows to manually enter any other estimates of the probabilities of being positive or negative, which may be estimated by other methods. Among other functionalities, the **sMSROC** package computes the AUC with confidence intervals, and provides plots for the ROC curve estimates, the predictive models, and the evolution of the AUCs across the follow-up time, providing different options for customizing the final graphics.

The remainder of the paper is organized as follows. In the Section 2, we present the sMS estimator and review its main properties. We provide a general insight of the structure of the **sMSROC** package in the Section 3. In Section 4 the main functions are described in detail. Two real-world datasets are used to illustrate the use of these functions in the diagnosis and prognosis scenarios. In the Section 5, we present a third real-world example, and show how the package can be used to assess the prognostic ability of a biomarker when data are interval censored. We want to present a disclaimer that the examples that the analyses provided here are used to demonstrate the use of the package only and they should not be used to inform any clinical decisions. Finally, we end in the Section 6 with a discussion of the potential uses of the sMSROC package.

2 The two-stage mixed subjects receiver operating-characteristic curve estimator

We first introduce the notation that will be used along this paper. Let X be a continuous random variable, with Cumulative Distribution Function (CDF) $H(\cdot)$, which models the behavior of the biomarker values. Let D be the binary random variable representing the event of interest, taking, without lost of generality, values 0 and 1, identifying negative and positive individuals, respectively. For prognosis scenarios, let T be the involved time-to-event random variable and let our aim be to predict the occurrence of the event of interest before a fixed point of time t. The binary variable depicting this event is given by D_t, which again takes the values 0, when $T > t$ (negative individuals) and 1, when $T \leq t$ (positive individuals). For sake of simplicity, we will remove the subscript t and will use the same notation in both scenarios.

The expression of the sensitivity can be written as

$$\begin{aligned} Se(c) =& \mathcal{P}(X > c \mid D) \\ =& \frac{\mathcal{P}(X > c,\, D)}{\mathcal{P}(D)} \\ =& \frac{\mathbb{E}_X[\mathcal{P}(X > c,\, D \mid X = x)]}{\mathbb{E}_X[\mathcal{P}(D \mid X = x)]} \\ =& \frac{\mathbb{E}_X[\mathbb{I}_{(c,\infty)}(x) \cdot \mathcal{P}(D \mid x)]}{\mathbb{E}_X[\mathcal{P}(D \mid x)]} \\ =& \frac{\int [\mathbb{I}_{(c,\infty)}(x) \cdot \mathcal{P}(D \mid x)] dH(x)}{\int (\mathcal{P}(D \mid x)) dH(x)}, \end{aligned} \tag{2}$$

where $c \in \mathbb{R}$; $\mathbb{I}_A(x)$ depicts the indicator function; D stands for the positive outcome ($D = 1$), and $\mathcal{P}(D \mid x) = \mathcal{P}(D \mid X = x)$.

Similarly, the specificity has the expression:

$$\begin{aligned} Sp(c) =& \mathcal{P}(X \leq c \,|\, \overline{D}) \\ =& \frac{\mathcal{P}(X \leq c,\, \overline{D})}{\mathcal{P}(\overline{D})} \\ =& \frac{\mathbb{E}_X[\mathcal{P}(X \leq c,\, \overline{D} \,|\, X = x)]}{\mathbb{E}_X[\mathcal{P}(\overline{D} \,|\, X = x)]} \\ =& \frac{\mathbb{E}_X[\mathbb{I}_{(-\infty,c]}(x) \cdot (1 - \mathcal{P}(D \,|\, x))]}{\mathbb{E}_X[(1 - \mathcal{P}(D \,|\, x))]} \\ =& \frac{\int [\mathbb{I}_{(-\infty,c]}(x) \cdot (1 - \mathcal{P}(D \,|\, x))] dH(x)}{\int (1 - \mathcal{P}(D \,|\, x)) dH(x)}, \end{aligned} \tag{3}$$

where $\overline{D}$ depicts the negative outcome ($D = 0$).

Let $(\mathcal{X}_N, \mathcal{D}_N) = \{(x_1, d_1), \cdots, (x_N, d_N)\}$ be an independent random sample where, for the i-th individual ($1 \leq i \leq N$), x_i is the biomarker value and d_i reports some information regarding the outcome of interest. Such information, in the diagnosis scenario, may provide the actual status of the individual ($d_i = \delta_i$, where $\delta_i = 0$ for a negative individual, or $\delta_i = 1$, for a positive one) or may be missing. When dealing with a time-dependent outcome, this information can also include the event/censoring time, according to the censorship pattern. That is, in the case of right censorship, $d_i = \{\delta_i, z_i\}$, being $z_i = \min\{t_i, c_i\}$, where t_i and c_i stand for the event and censoring times, respectively. Under interval censorship, $d_i = \{l_i, r_i\}$, where l_i and r_i are the lower and upper bounds of the observed interval containing the event time ($l_i \leq t_i \leq r_i$). The sensitivity and specificity given in (2) and (3) can be estimated through:

$$\widehat{Se}(c) = \frac{\sum_{i=1}^{N} I_{(c,\infty)}(x_i) \cdot \widehat{\mathcal{P}}_N(D \,|\, x_i)}{\sum_{i=1}^{N} \widehat{\mathcal{P}}_N(D \,|\, x_i)}, \tag{4}$$

$$\widehat{Sp}(c) = \frac{\sum_{i=1}^{N} I_{(-\infty,c]}(x_i) \cdot (1 - \widehat{\mathcal{P}}_N(D \,|\, x_i))}{\sum_{i=1}^{N} (1 - \widehat{\mathcal{P}}_N(D \,|\, x_i))}, \quad c \in \mathbb{R}, \tag{5}$$

where $\widehat{\mathcal{P}}_N(D \,|\, x)$ is chosen to be an adequate estimator of $\mathcal{P}(D \,|\, x)$.

Plugging-in the expressions (4) and (5) in the definition of the ROC curve given in (1), we obtain the **Two-stage Mixed Subject (sMS) ROC curve estimator**, to which we will refer as sMS estimator:

$$\widehat{\mathcal{R}}(u) = \widehat{Se}\left([1 - \widehat{Sp}]^{-1}(u)\right), \quad u \in [0, 1],$$

where $\widehat{Sp}^{-1}(\cdot) = \inf\{x : \widehat{Sp}(x) \geq \cdot\}$.

We briefly review some features of the sMS estimator already introduced in Díaz-Coto et al. (2021):

- The relationship between the biomaker and the outcome is modeled by $\mathcal{P}(D \,|\, x)$ (the predictive model). In the first stage, the sMS estimator approximates the predictive model through the most suitable probabilistic model (e.g. proportional hazards, logistic regression). In the second stage, the rest of the unknown parameters is estimated by the corresponding empirical estimators. The first stage is specially important because the performance of the sMS estimator is highly dependent on the fit of the predictive model to the actual relationship between the biomarker and the outcome.
- The sMS estimator does not need to consider the individuals as fully positive or fully negative. Each individual can be modeled as mixed: partially positive and partially negative (hence the name "mixed subjects"). The weight allocated to each possibility is determined by the predictive model considering the biomarker value in the specific individual.
- The sMS estimator can handle missing values in the outcome as well as censored observations (latter frequently associated with prognosis studies). The individuals with missing outcome are supposed to be missing at random (MAR) however; that is, their characteristics in the sample should be similar to those with complete information. Under this assumption, their potential outcome is determined by the predictive model for the particular biomarker value.
- The sMS estimator generalizes some of the ROC and C/D ROC curve existing estimators. In the simplest diagnosis scenario, where the real status of all individuals is known, we can estimate the predictive model $\mathcal{P}(D \,|\, x)$ through the average of the status of those having a biomarker value of x. The resulting estimator would be the well-known empirical ROC curve estimator

(Hsieh and Turnbull, 1996). In the prognosis scenario, considering the adequate estimators for the predictive model, it is clear the connection with the C/D ROC curve estimators under right censorship proposed in Martínez-Camblor et al. (2016) and Li et al. (2018), and with the estimator proposed in Díaz-Coto et al. (2020a) under interval censorship. For particular parametrizations, the sMS estimator can as well be the C/D ROC curve estimator proposed by Chambles and Diao (2006) and by Song and Zhou (2008).

- Under certain conditions both the sMS estimator and its corresponding estimator for the AUC are asymptotically normal distributed. We provide two approximations for the variance of the AUC estimator: the Theoretical Variance Estimation (TVE), based on a theoretical expression, and the Empirical Variance Estimator (EVE), which avoids dealing with the expression of the variance of the predictive model. Explicit expressions for both the TVE and the EVE approximations are provided in the Appendix of this manuscript. Reported confidence intervals are $\mathcal{A} \pm \lambda_\alpha \cdot \hat{\sigma}$, with $\mathcal{A}$ and $\hat{\sigma}^2$ the AUC and variance approximations, respectively, and λ_α the adequate quantile based on the normal distribution.

It is worth clarifying that, although related, the sMS estimator is not a single imputation procedure. In this sense, we are not considering here the presence of missing data in the biomarker. The goal of the sMS estimator is not to impute the unknown values of the outcome, but to use the estimated probabilities to approximate the ROC curve. These probabilities could be used even for those subjects for whom we already know the actual status. For instance, in a standard study in which we collect the status of each single participant, the empirical model would be an extreme situation in which the probability of being positive is determined by the actual observed status of the subject (probability 1 or 0). As we have already noted, in this case, the resulting sMS estimator would be the empirical ROC curve estimator. However, we could model these probabilities by the standard binary logistic regression to obtain a smoothed ROC curve estimate. For sure, the quality of this estimation would depend on the goodness of fit of the regression model. For time-to-event outcomes, we can use the actually observed follow-up times for computing the probabilities. Notice that, if the target of interest is to predict events prior to the point $t = 5$, participants who still alive at point 4.99 are more likely to be alive at 5 than those censored at 0.01. One of the main advantage of the sMS estimator is its flexibility, which allows to adapt the procedure to several types of data, including different censoring models, and provides a variety of techniques under the same umbrella.

3 An overview of the package

The main goal of the **sMSROC** package (available at `https://CRAN.R-project.org/package=sMSROC`) is to compute the sMS estimator and related elements, which support the assessment of the diagnostic/prognostic ability of continuous biomarkers. Since `R` programming is mostly based on objects (López-Ratón et al., 2014), the **sMSROC** package consists in a set of functions performing specific tasks.

Table 1 provides a summary of these functions, grouped by their common features. The functions have been classified as primary and secondary. Among the former, we consider those directly run by the end-user to perform the exploratory data analysis, compute the sMS estimator and other metrics (such as the AUC and its confidence interval), and to summarize the computed results. We refer to the rest of the functions as secondary, as these are mainly called by other functions and not meant to be used by the end-user, primarily. We will describe the primary functions in more detail in the next sections.

The **sMSROC** package uses some functionalities already implemented in other packages. In a non-exhaustive list we highlight: the functions `Surv` and `ic_sp`, from the **survival** (Terry M. Therneau and Patricia M. Grambsch, 2000) and **icenReg** (Anderson-Bergman, 2017) packages, which provide estimates of the survival function under right and interval censorship, respectively; the `rcs` function, from the **rsm** package (Harrell Jr, 2023), that computes the cubic splines approximation; the `%dopar%` function from the package **foreach** (Microsoft and Weston, 2022), used to perform parallel computing; the `flextable` function, from the package with the same name (Gohel and Skintzos, 2023), which provides formatted outputs for the tables and the `ggplot` and `plotROC` functions, from the packages **ggplot2** (Wickham, 2016) and **plotROC** (Sachs, 2017), used to obtain well-formatted and interactive final plots, respectively.

Table 1: Functions included in the sMSROC package grouped by the similarity of the tasks performed.

Main Functionality	**Functions**	**General Description**
Exploratory data analysis	`explore_table` `explore_plot`	Perform a descriptive analysis of the biomarker values on the different samples
sMS ROC curve estimator	`sMSROC`	Core function that is actually a wrapper of those functions computing each element related to the sMS estimator

Check-ups	`check_type_outcome`
	`check_conf_int`
	`check_grid`
	`check_marker_binout`
	`check_marker_timerc`
	`check_marker_timeic`
	`check_tim`
	`check_meth`
	`check_probs`
	`check_ncpus`
	`check_nboost`
	`check_ci_cl`
		Verify the integrity and the consistency of the parameters of the functions entered by the end-users

sMS ROC	`sMS_binout`
	`sMS_timerc`
	`sMS_timeic`
		Compute the sMS ROC curve estimates in each particular scenario

Predictive models	`pred_model_binout`
	`pred_model_timerc`
	`pred_model_timeic`
	`pred_model_emp`
		Compute the estimates for the predictive models according to specific probabilistic models (first stage of the sMS estimator)

ROC curve	`compute_ROC`	Computes the ROC curve and the AUC through the estimators of sensitivity and specificity (second stage of the sMS estimator)

Confidence intervals for the AUC	`auc_ci_boot`
	`auc_ci_emp`
	`auc_ci_var`
		Compute confidence intervals for the AUC according to the selected method

Plots	`sMSROC_plot`
	`evol_AUC`
	`prob_pred`
		Plot the sMS ROC estimate, the evolution of the AUCs, and the predicted probabilities

Print	`conf_int_print`	Prints certain components of the sMS estimate

4 Primary functions

4.1 Exploratory data analysis

The exploratory analysis of the data is carried out by the `explore_table` and `explore_plot` functions. They allow to have an insight of the distribution of the biomarker on positive and negative individuals and on those whose belonging group is unknown. This may help to the selection of the most suitable predictive model for each particular problem. Both functions share the input parameters collecting the sample information that was formally introduced previously: the biomarker values and the information regarding the outcome of interest, which varies depending on the scenario. They also have specific parameters according to the performed task. The functions provide numerical and graphical outputs.

The function `explore_table` computes the most common descriptive statistics for the pooled sample and the samples of the different types of individuals. The input parameters are:

- **marker** a vector of the biomarker values.
- **status** a numeric vector with the status of the individuals. The highest value represents the event of interest. The lowest value represents the absence of the event of interest. All other values are ignored.
- **observed.time** a vector with the observed times for each subject (prognosis scenario under right censorship). Notice that these values may be the event times or the censoring times.
- **left** a vector with the lower bounds of the observed intervals. It is mandatory, when computing the sMS estimator for assessing prognostic biomarkers under interval censorship. It will be ignored in other situations.
- **right** a vector with the upper bounds of the observed intervals. Like the previous parameter, it is mandatory in the prognosis scenario under interval censorship and ignored in other situations. Non available, NA and ∞ ('Inf') are admissible values to indicate that the event of interest did not occur prior to the last observation time.
- **time** point of time at which the time-dependent sMS estimator will be computed. The default value is 1. This parameter is mandatory in the prognosis scenario.
- **d** number of decimal positions to which all results will be rounded. The default value is 2.
- **...** rest of the parameters supplied to the flextable function. These can be used to customize the output table as desired.

In diagnosis scenarios, it is clear when individuals are either positive or negative. When dealing with time-dependent outcomes, this status depends on a fixed point time t at which they are evaluated. Particularly, in the interval censorship case, if the last revision time in which the event had already happened took place before the set time t, the individual is positive at t. If there exists a revision time beyond t and the event has not been observed yet, the individual is negative at t. When the event occurs between two consecutive revision times containing the set time t, nothing is known about the status of the individuals at t, because it is not actually observed when the event happened. We refer to these individuals as *undefined* or *mixed*.

The consistency of the incoming parameters is verified by secondary functions. Next, the type of scenario handled (diagnosis/prognosis, under right or interval censorship) is determined. The functions `explore_table`, `explore_plot` and `sMSROC` share both steps.

The output of the function is a list with two components:

- **summary** a matrix whose columns are the name of the groups, their size and the descriptive statistics: minimum, maximum, mean, standard deviation and first, second and third quartiles. The rows show the results for positive, negative, missing/censored/undefined individuals and the pooled sample.
- **table** an object of class **flextable** that collects the descriptive information from the previous matrix in a table, which can be customized according to the preferences of the users by the entered parameters.

The function `explore_plot` plots the kernel density estimations for the biomarker within both the positive and the negative individuals. The input parameters are those related to the sample information described for the `explore_table` function.

The output is a list with three components:

- **plot** an object of class **ggplot** with the density functions of the biomarker on the positive and the negative individuals. The user can add layers to customize the final plot according to the rules of the **ggplot2** package.

- **neg** a vector with the marker values of the negative individuals.
- **pos** a vector with the marker values of the positive individuals.

Example 1 **[Exploratory data analysis]: the diabetes dataset**

We first consider the study of the ability of *stabilized glucose* to diagnose diabetes (defined through a subsidiary measure: a value of glycosylated hemoglobin greater than 7.0) in an African-American population of central Virginia (USA). We consider the dataset freely available at `https://hbiostat.org/data/`. The subset data **diabet**, used here, is delivered as part of the **sMSROC** package. More information about this study can be found in Willems et al. (1997).

A total of 60 individuals out of the 403 included were diabetic (positive), and 330 were classified as non-diabetic (negative). Besides, there were 13 individuals without glycosylated hemoglobin value, so we cannot determine their actual status (undefined). The next piece of code provides the distribution of the *stabilized glucose* on these groups, shown in HTML format in Table 2:

```
> library(sMSROC)
> data(diabet)
> expl <- explore_table(marker=diabet$stab.glu, status=diabet$diab)
> expl$table
```

Table 2: Object of class **flextable**, one of the elements of the output list of the `explore_table` function. The usual descriptive statistics of the *stabilized glucose* biomarker on the total sample and the samples of Positive, Negative and Undefined observations are shown.

Sample	Size	Minimun	Maximun	Mean	Sd	Variance	Q1	Median	Q3
Positive	60	60	385	194.17	77.44	5,996.68	120	186.0	241.25
Negative	330	48	371	91.55	26.87	721.77	79	86.5	97.00
Miss/Cens/Und	13	68	105	86.69	10.26	105.23	80	84.0	88.00
Total	403	48	385	106.67	53.08	2,817.13	81	89.0	106.00

Left panel of Figure 1 shows the kernel density estimations of the *stabilized glucose* on positive and negative individuals generated through the code:

```
> library(ggplot2)
> density <- explore_plot(marker=diabet$stab.glu, status=diabet$diab)
> output  <- density$plot + xlab("Stabilized Glucose") +
                          scale_x_continuous(breaks = seq(0, 400, 50),
                                             labels = seq(0, 400, 50),
                                             limits = c(0, 400))
> output
```

4.2 Main function

sMSROC is the main function in the package. It computes the sMS ROC curve estimator and its associated AUC with confidence intervals, estimated using the bootstrap percentile (BP) method, and according the approximations EVE and TVE given in Section 2. The function has the following input parameters:

- **meth** method for approximating the predictive model $\mathcal{P}(D|x)$. There are several options available:
  - `E` implements the naive method where missing, censored and undefined individuals are removed from the data.
  - `L` in the diagnosis scenario this option models the probability of being positive as:

$$\mathcal{P}(D|x) = \frac{1}{1+\exp\{-(\beta_0+\beta_1\cdot x)\}}, \quad \beta_0,\beta_1 \in \mathbb{R}.$$

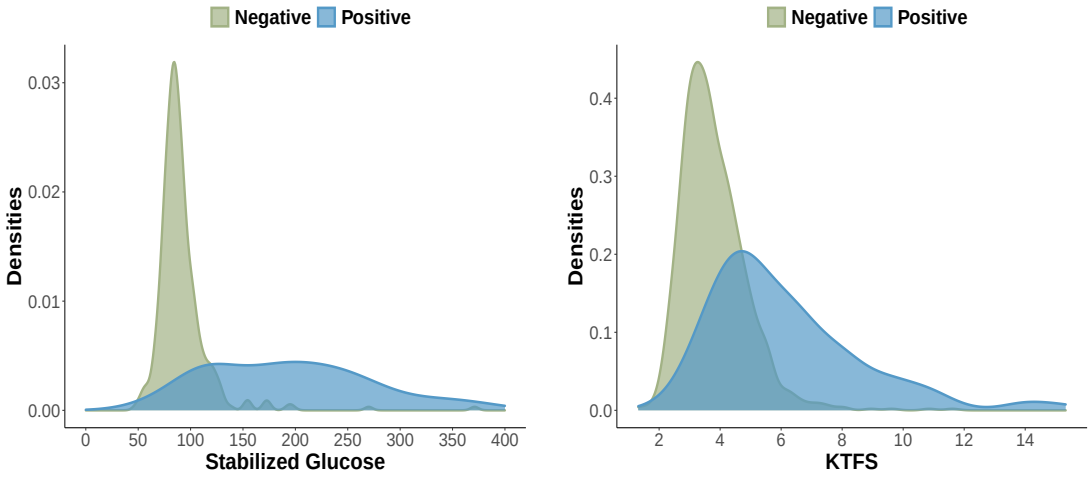

Figure 1: Left: Kernel density estimations of the *stabilized glucose* biomarker on negative and positive individuals. Right: kernel density estimations of the *KTFS* biomarker on negative and positive individuals after 5 years from transplantation.

In the case of prognosis scenarios under right censorship, a proportional hazards Cox regression model is used to approximate $\mathcal{P}(T \leq t|x)$ (and therefore, $\mathcal{P}(D|x)$).

$$\mathcal{P}(T \leq t|x) = 1 - \exp\{-\Gamma_0(t) \cdot \exp\{\beta \cdot x\}\},$$

where $\Gamma_0(\cdot)$ stands for the so-called cumulative baseline hazard function (Cox, 1972). Under interval censorship, $\mathcal{P}(T \leq t|x)$ is approximated through the model proposed in Díaz-Coto et al. (2020a):

$$\mathcal{P}(T \leq t|x) = \frac{S(U|x) - S(t|x)}{S(U|x) - S(V|x)}, \tag{6}$$

where $U = \min\{t, L\}$ and $V = \max\{t, R\}$, being L and R the random variables depicting the edges of the observable interval $(L, R]$. The expression $S(t|x)$ stands for $S(t|X = x)$, that is, the survival function at t, given the marker value. It is estimated by proportional hazards model under interval censorship (Finkelstein, 1986), applying linear interpolation inside the Turnbull intervals (Turnbull, 1976).

In this case, the parameter **all** indicates whether this approximation applied to all individuals or just to the mixed/censored/undefined ones.

- `S` approximates, in diagnosis scenarios, the logit transformation of the predictive model via a cubic spline function. That is:

$$\mathcal{P}(D|x) = \frac{1}{1 + \exp\{-s(x)\}},$$

where $s(\cdot)$ depicts some smooth function, estimated from restricted cubic splines (Harrel, 2015). In the prognosis scenario under right censorship, we consider a proportional hazards model with a more flexible option to approximate the predictive model:

$$\mathcal{P}(T \leq t|x) = 1 - \exp\{-\Gamma_0(t) \cdot \exp\{s(x)\}\},$$

where a penalized splines procedure (Hurvich et al., 1998) estimates the smooth function $s(\cdot)$. Under interval censorship the predictive model is obtained by

$$\mathcal{P}(T \leq t|x) = 1 - S(t|x),$$

where $S(t|x)$ is estimated as before through a proportional hazards model under interval censorship.

- **probs** a vector of manually entered predicted probabilities. This argument is useful if the user wants to estimate the predicted probabilities via a different model than the ones currently offered by the package. In this case, the argument **meth** will be ignored.

- **sd.probs** a vector with the standard deviation of the probabilities entered as **probs**. It is an optional parameter.
- **grid** grid size for computing the ROC curve estimate. The default value is 1000. It is also used to compute the AUC.
- **conf.int** argument with two possible values indicating whether confidence intervals for the AUC will be computed (`T`) or not (`F`).
- **ci.cl** confidence level at which the confidence intervals for the AUC will be calculated. The default value is 95%.
- **ci.meth** method for computing the AUC confidence intervals. There are three options available:
  - `V` method that uses the TVE variance approximation.
  - `E` method which uses the EVE approximation.
  - `B` confidence intervals based on BP.
- **ci.nboost** number of bootstrap samples to be run when the option `B` is chosen as **ci.meth** parameter. The default value is 500.
- **parallel** argument with two possible values which indicates whether parallel computing will be carried out (`T`) or not (`F`). There are two processes that currently support parallel computing: the `B` and `V` options to obtain confidence intervals for the AUC.
- **ncpus** number of CPUS that will be used when parallel computing is chosen.
- **all** parameter indicating whether all probabilities given by the predictive model should be considered (`T`) or only those corresponding to individuals whose condition as positive or negative is unknown (`F`).

The sMSROC function returns an object of class **sMSROC**. It is a list of the following elements:

- **thres** a vector of thresholds at which the sensitivity and the specificity have been computed.
- **SE** a vector with the sensitivities at the considered thresholds **thres**.
- **SP** a vector with the specificities calculated at the thresholds **thres**.
- **probs** a vector with the predictive model estimates for each threshold. Recall that they represent the probability for the observations of being positive, given their marker values.
- **u** sequence of points at which the sMS estimator will be computed. Its length is determined by the **grid** selection parameter.
- **ROC** sMS ROC curve estimate computed at each value of the sequence **u**.
- **time** the point of time at which the ROC curve has been computed, in the case of time-dependent outcomes.
- **auc** area under the sMS ROC curve estimate. It is computed as the sum of the area of the rectangles whose base lies on two consecutive values of the sequence **u** and whose height is the sMS estimate value that corresponds to the lower edge of the base.
- **auc.ci.cl** confidence level at which the confidence interval for the AUC has been computed.
- **auc.ci.l** lower bound of the confidence interval for the AUC.
- **auc.ci.u** upper bound of the confidence interval for the AUC.
- **ci.meth** method used for estimating the confidence intervals for the AUC.
- **data** a list whose elements are, in addition to the type of outcome handled, the set of parameters that had been used to compute the sMS estimator. The values for the **grid**, **meth**, **parallel** and **ncpus** elements are the default or those entered by the users. The **marker**, **status**, **observed time**, **left** and **right** vectors do not contain the positions that correspond to the missing marker values. The **outcome** element is a vector taking the value 1 for the positive individuals; 0, for negative and -1 for the missing/censored/undefined ones.
- **message** a table with the warning messages generated along the computation process.

The computation of the sMS ROC curve estimator is wrapped in three secondary functions according to each scenario. They have the same structure: a part in which the predictive model is estimated (first stage of the sMS estimator), and a common part where the remainder unknown elements are approximated through their empirical estimators (second stage of the sMS estimator).

The functions computing the predictive models have a common output: a list with the ordered marker values and their corresponding estimated probabilities. These two components and the **grid** are the input parameters of the `compute_ROC` function, which implements the computation of the

second stage of the sMS estimator. This function provides: the sMS estimates for the sensitivity and the specificity according to the expressions given in (4) and (5); the ROC curve approximation obtained from these estimates (at the granularity level chosen through the **grid** parameter) and the underlying AUC. It is possible as well to directly enter the probabilities corresponding to the predictive model as parameter **probs**. In this case, none of the functions that compute the predictive model are called.

When the **conf.int** parameter is set to `T`, confidence intervals for the AUC are provided at the confidence level indicated as **ci.cl** parameter. They are computed according to the method entered as **ci.meth** parameter.

- When the chosen method is `B` the function `auc_ci_boot` is called and the confidence intervals based on BP are calculated. Depending on the type of outcome handled, the corresponding functions for computing the AUC under the sMS estimator are called **ci.nboost** times. Since the bootstrap processes can be time-consuming, the function can be run in parallel via the argument **parallel**. In this case, the number of desired CPUs to be used should be indicated through the parameter **ncpus**. The package **foreach** was used to implement the parallel computation.
- Placing the option `V` as **ci.method**, the function `auc_ci_nvar` is called and the confidence intervals are computed according to the asymptotic normality of the AUC estimator, based on the TVE approximation. The expression for the variance in this method depends on the variance of the predictive model estimates. That variance is calculated in an independent auxiliary function by bootstrapping (it is possible as well to perform parallel computation to carry out this task). When the probabilities of the predictive model have been directly entered by the users, there is also the option of indicating, manually, the corresponding standard deviation for these probabilities.
- When the selected method is `E`, the function `auc_ci_emp` uses the asymptotic normality of the AUC estimator, in this case, the variance is calculated through the EVE approximation.

The three functions have the same output: a list with two components, the lower and upper edges of the computed confidence intervals for the AUC.

Example 1 [ROC curve models]: the diabetes dataset

Coming back to the diabetes dataset, we compute the ROC curves to assess the ability of *Stabilized glucose* to identify patients with diabetes. First, we only include this biomarker in the model, and since we observed that its distribution within the positive and the negative populations differs in location, spread, and shape, we used a smooth logistic regression for the first-stage estimation.

```
> roc_diabetes <- sMSROC(marker=diabet$stab.glu, status=diabet$diab,  meth="S")
> roc_diabetes

The AUC is 0.926.
Predictive model computed through a smooth logistic regression model, based on 60 positive,
330 negative, and 13 undefined (mixed) subjects.
```

The object `roc_diabetes` also contains the following components

```
> summary(roc_diabetes)

        Length Class  Mode
thres    403   -none- numeric
SE       403   -none- numeric
SP       403   -none- numeric
probs    403   -none- numeric
u       1001   -none- numeric
ROC     1001   -none- numeric
auc        1   -none- numeric
ci.meth    3   -none- character
data       6   -none- list
message    4   -none- character
```

which allow to perform a number of customized figures and analyses, including the selection of thresholds under different criteria.

On the other hand, if we want to use an alternative model in the first-stage, for instance, by including additional information which could help us to have a more accurate prediction of the real

status of the undefined subjects, we just have to perform this first-stage out of the sMSROC function, and save the vector of the predicted probabilities. This is illustrated in the following chunk of code, where we use logistic regression with first-order effects for both the biomarker and age.

```
> alt_mod <- glm(diab ~ diabet$stab.glu + diabet$age, family = 'binomial')
> prob_model <- predict(alt_mod, type = 'response',
                        newdata = data.frame(diabet$diab, diabet$stab.glu, diabet$age))
```

Then, we include these probabilities in the function

```
> roc_diabetes_prob <- sMSROC(marker=diabet$stab.glu, status=diabet$diab,
                              probs=prob_model)
> roc_diabetes_prob

The AUC is 0.886.
Predictive model externally computed. Based on 0 positive, 0 negative,
and 403 undefined (mixed) subjects.
```

Notice that, in this example, all the subjects are considered as mixed, since none of them are considered as fully positive not fully negative. If we would want to apply this model only on those subjects for which the actual status is unknown, we should introduce probabilities of 1 of 0 for the actually positive, and actually negative, respectively.

Example 2 [ROC curve models]: the kidney transplant failure score (KTFS) dataset

In this second example, our aim is to evaluate the prognostic ability of the *Kidney Transplant Failure Score* (KTFS) to predict the graft failure after five years from kidney transplantation. The KTFS is a composite score build on the base of accepted risk factors of graft loss (Foucher et al., 2010). We will use a subset of the DIVAT cohort (`https://www.divat.fr`) delivered at the **RISCA** package, and now also included in our package. This dataset, **ktfs**, contains the follow-up time from transplantation in years to either of the graft failure or the censoring time (in many cases due to death), a graft failure indicator, and the KTFS score for 2,169 kidney transplant recipients. The distribution of the KTFS score on both patients with graft failure within 5 years (108), and those with a functional graft after 5 years (954) was depicted in Figure 1 (right). Notice that 1107 patients were undefined (follow-up below 5 years and graft working).

```
> data(ktfs)
> roc_KTFS <-  sMSROC(marker=ktfs$score, status=ktfs$failure,
>              observed.time=ktfs$time, time=5, meth="L",
>              conf.int="T", ci.meth ="E")

The AUC is 0.763.
Predictive model computed through a Cox PH regression model, based on 108 positive, 954 negative,
and 1107 undefined (mixed) subjects.
```

4.3 Summarize and plot functions

The **sMSROC** package includes functions which provide numerical and graphical summaries of the data contained in the object returned by the *sMSROC* function. We describe them below.

The `sMSROC_plot` provides informative plots of the sMS ROC curve estimate. The function has the following input parameters:

- **sMS** an object of class **sMSROC**.
- **m.value** marker value. When specified, the point which corresponds to that marker value is added over the plot of the ROC curve.

The function generates two different types of graphics. On one hand, it computes a basic plot approximating the ROC curve by the pairs given by the sequences **1 - SP** and **SE**, from the **sMSROC** object. We have added to this plot the layers `geom_roc()` and `roc_style()` from the **plotROC** package, to obtain a final object that could take advantage of the whole functionality of this package. On the other hand, we produce a customized graphic of the ROC curve whose class is **ggplot** by plotting the sequence **1 - SP** against **SE**. In the case that a number of **m.value** is indicated, the final plot displays over the ROC curve line the point that corresponds to the entered value.

The output of the function is a list with two components:

- **basic.plot** an object that can be used and customized using the tools from the **plotROC** package.
- **roc.plot** an object of class **ggplot**. Although it is already customized (e.g. title, colors or axis labels) the users can make their own changes by adding the corresponding layers through the tools available in the **ggplot2** package.

The function `evol_AUC` provides a graphic with the areas under the time-dependent ROC curves computed by the sMS estimator over a sequence of times. Its input parameters include the sample information for time-dependent outcomes (**marker**, **status**, **observed.time**, **left** and **right**); the **time**, that in this case is a vector containing the points of time at which the AUC will be computed; the method of computation (**meth**) and the **grid**. The features of these parameters are the same described for the corresponding functions in the package. The function `evol_AUC` calls `sMSROC` at each of the times indicated in the vector **time**, and the AUC is computed according to the parameters indicated. The output is a list with three elements:

- **evol.auc** an object of class **ggplot**. It is a graphic line plotting the AUCs at the considered times.
- **time** a vector with the values of the **time** entered as parameter.
- **auc** a vector with the values of the AUCs computed at the times indicated at the **time** parameter.

The function `pred_probs` plots the predicted probabilities estimated from the predictive model for each of the marker values. It may provide a 95% pointwise confidence intervals. The input parameters of the function are:

- **sMS** an object of class **sMSROC**.
- **var** argument with two possible values indicating whether the pointwise confidence intervals should be computed (`T`) or not (`F`).
- **nboost** number of bootstrap samples for computing the pointwise confidence interval.
- **parallel** argument indicating whether parallel computing will be carried out (`T`) or not (`F`).
- **ncpus** number of CPUS to be used in the case of choosing parallel computing.

The function `pred_probs` generates a graphic for the probability estimation of the predictive model versus the marker values. As usual, this is a **ggplot** object which can be customized by the user. In the case that the **var** option is set to `T`, the function computes and plots 95% pointwise confidence intervals on the same graphic. The variance of the probability estimates is computed via bootstrap. The output of the function is a list with four components:

- **plot** an object of class **ggplot**.
- **thres** a vector of marker values (x-axis coordinates).
- **probs** a vector containing the predicted probabilities (y-axis coordinates).
- **sd.probs** a vector containing the estimation of the deviation of the predicted probabilities.

Example 2 [ROC curve plots]: the kidney transplant failure score (KTFS) dataset

The next piece of code returns the plot of the ROC curve computed on the data from the *KTFS* example (top-left). We only show the basic plot, however, it can be customized with elements from the **plotROC** package. The code also generates the probabilities derived from the predicitive model used in the first stage (i.e. proportional hazard Cox regression), and included in the same panel (top-right), and the evolution of the AUC over ten years from kidney transplantation (bottom).

```
> # ROC curve
> plot_KTFS <- sMSROC_plot(sMS = roc_KTFS, m.value = 3.9)
> plot_KTFS$rocplot
>
> # Evolution of the AUCs
> aucs <- evol_auc(marker = ktfs$score, status = ktfs$failure,
>                  observed.time = ktfs$time,
>                  time = seq(2, 10), meth = "L")
> plot_aucs <- aucs$evol.auc +
>              scale_x_continuous(limit = c(2, 10),
>                                 breaks = seq(2, 10, 1)) +
>              scale_y_continuous(limit = c(0.4, 1),
>                                 breaks = seq(0.4, 1, 0.1))
> df1 <- data.frame(x = c(2,10), y = c(0.5, 0.5))
```

```
> plot_aucs <- plot_aucs +
> geom_line(data = df1, aes(x, y), linewidth = 0.9, colour = "gray", linetype = "twodash")
> plot_aucs
>
> # Predictive model
> probs <- probs_pred(roc_KTFS, var = "T")
> plot_probs_pred <- probs$plot + xlab("KTFS")
> plot_probs_pred
```

All these plots are arranged in the Figure 2.

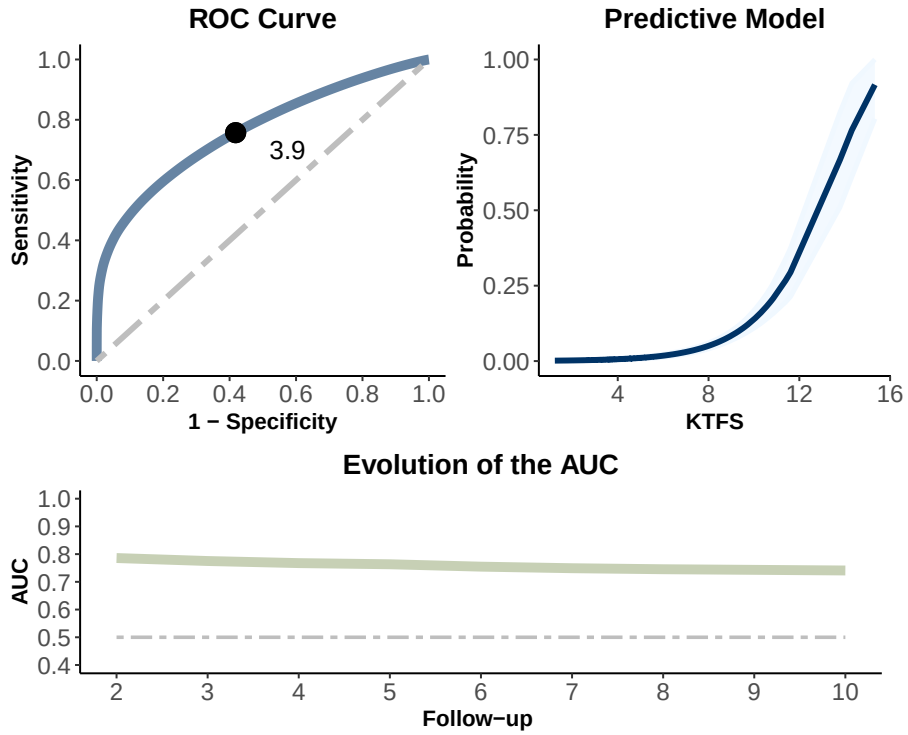

Figure 2: Upper left: graphic of ROC curve estimate obtained from the `sMSROC_plot` function, customized to show the point of the curve corresponding to a given *KTFS* value. Upper right: estimated probabilities from the predictive model with 95% pointwise confidence intervals computed for the biomarker. Bottom: evolution of the AUCs over 10 years from kidney transplantation for the *KTFS* score.

Finally, the function `conf_int_print` prints the values, method of computation, and the confidence level of the confidence intervals calculated for the AUC. Since the **sMSROC** object may contain information stored in large list of components, we only print a single summary, such as the lower and the upper bounds of the confidence intervals, the level at which they were obtained, and the method used for their computation. The input of this function is an object of class **sMSROC** and the output is a string including the described information. As example, below is shown the code for printing the AUC and 95% confidence intervals for the *KTFS* at 5 years:

```
> conf.int.print(roc_KTFS)
"AUC: 0.76; 0.95\% C.I.[0.6, 0.93]"
```

5 Example 3: the fibrosis dataset

We finally consider the fibrosis dataset. This synthetic data set ships alongside singR and emulates a retrospective study carried out at three different medical centers in Spain. The goal was to determine the capacity of a score punctuation (based on the age, different polymorphisms, and other variables)

to predict the worsening of the fibrosis stage in patients with chronic Hepatitis C (HC). A total of 722 individuals infected by the HC virus, and underwent revision since a particular date were enrolled. The date of the diagnosis of HC and a number of variables determining the risk score such as the age, gender, alcohol consumption, and different polymorphism variants were also collected. Reader interested in having more information about the original study are referred to Vidal-Castiñeira et al. (2020).

Particularly, we are interested in knowing the prediction ability of the constructed risk score at 5, 10, and 15 years from the HC onset. However, for each patient, we only know whether or not the fibrosis worsened within the interval $(0, R_i]$ or (R_i, ∞) $(1 \leq i \leq 722)$, where R_i is the time between the diagnosis and the revision dates. Therefore, we have an interval censorship scenario in which, for instance, at 5 years, the ith patient is positive if they were positive in the revision and $R_i \leq 5$, negative if they was negative in the revision and $R_i \geq 5$, and undefined otherwise. The next piece of code deals with the ROC curve construction at 5 years. Since higher values of the score are associated with smaller probabilities of having the event, we have transformed the values adequately.

```
> data(fibrosis)
> explore_table(marker = -fibrosis$Score, left = fibrosis$Start,
>                 right = fibrosis$Stop, time = 5)$summary

         Sample Size Minimun Maximun   Mean   Sd Variance  Q1 Median Q3
1      Positive   21      -8      -2  -5.95 1.63     2.65  -7     -6 -5
2      Negative  112     -22      -3 -11.30 3.21    10.30 -13    -11 -9
3 Miss/Cens/Und  589     -21      -3 -10.02 3.09     9.56 -12    -10 -8
4         Total  722     -22      -2 -10.10 3.19    10.19 -12    -10 -8

> roc_fibrosis_5 <- sMSROC(marker = -fibrosis$Score, left = fibrosis$Start,
>                         right = fibrosis$Stop, meth = "L", time = 5)
> roc_fibrosis_5

The AUC is 0.647.
Predictive model computed through a D. Finkelstein PH regression model, accounting to
the length of the observed intervals, based on 21 positive, 112 negative
and 589 undefined (mixed) subjects.

> sMSROC_plot(roc_fibrosis_5)$rocplot
```

Figure 3 shows the ROC curves at 5, 10, and 15 years using the sMS ROC curve estimates (top-left), and the estimator proposed by Beyene and El Ghouch (2022) (top-right) and recently implemented in the package **cenROC**. Besides, since the object `roc_fibrosis_5` also contains the values of both the sensitivity and the specificity for each potential threshold, the next simple piece of code allows to compute the weighted Youden index (Martínez-Camblor, 2011).

$$J_\lambda = \max_{x \in \mathbb{R}} \{\lambda \cdot Se(x) + (1-\lambda) \cdot Sp(x)\} \quad \lambda \in [0,1],$$

and its associated threshold. Figure 3 (middle) depicts J_λ at 5 years, and highlights some of the thresholds. Notice that $J_{1/2}$ is equivalent to the Youden index. We also include the AUC evolution along the follow-up computed through the **sMSROC** (blue line) and **cenROC** (red line) packages. AUCs at 5, 10, and 15 years were 0.647, 0.680 and 0.687 for the sMS ROC curve (Figure 3 bottom), and 0.640, 0.653 and 0.680 for the cenROC-based estimations.

```
> lambda <- seq(0, 1, length = 101)
> Yw <- seq(0, 1, length = 101)
> Tw <- seq(0, 1 ,length = 101)

> for (j in 1:101) {
> Yw[j]<- max(lambda[j]*roc_fibrosis_5$SE + (1-lambda[j])*roc_fibrosis_5$SP)
> Tw[j]<- roc_fibrosis_5$thres[which.max(lambda[j]*roc_fibrosis_5$SE +
                                        (1-lambda[j])*roc_fibrosis_5$SP)]}
```

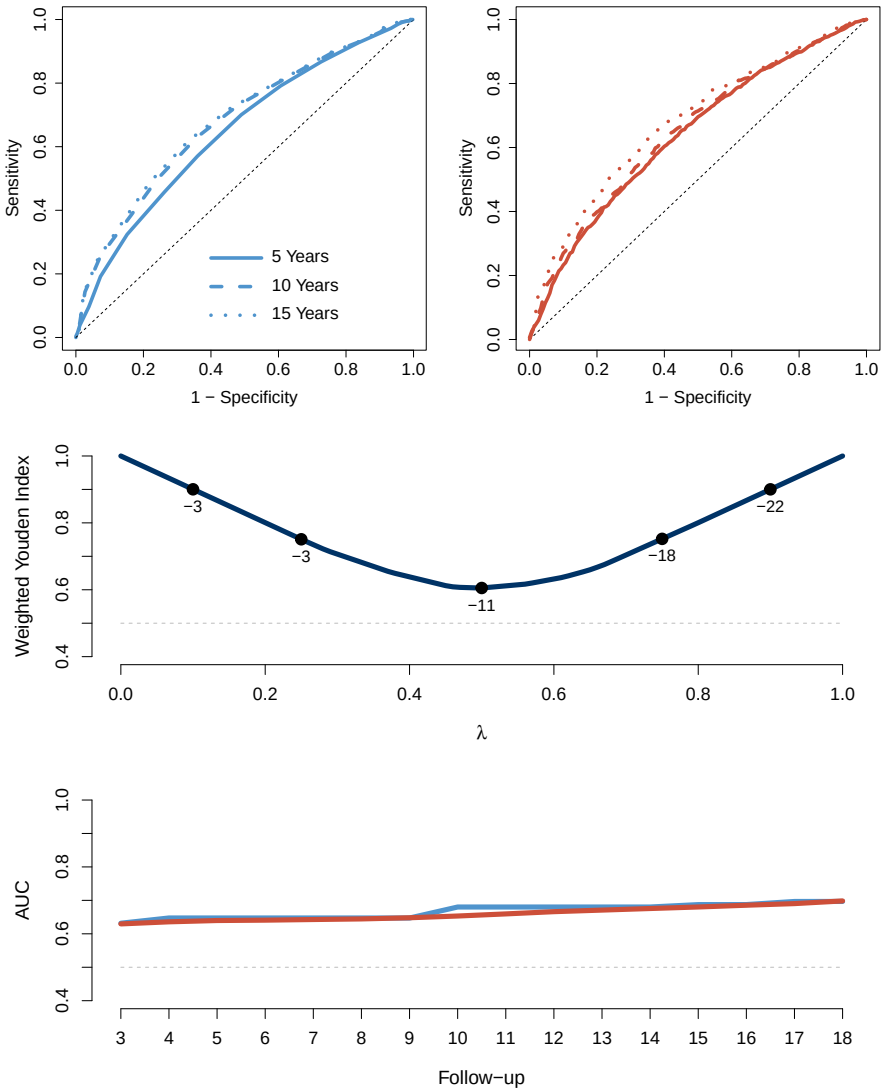

Figure 3: Top: ROC curves estimates at 5, 10 and 15 years from the **sMSROC** (left), and from the **cenROC** (right) packages. Middle: Weighted Youden Index at 5 years from the sMS ROC curve estimation. Bottom: AUC evolution from 3 to 18 years for the two considered estimators.

6 Conclusions

We presented the new R package **sMSROC** which implements the two-stage mixed-subjects ROC curve estimator. This procedure allows the user to assess the classification performance of both diagnostic and prognostic biomarkers. The package offers a set of exploratory functions which allow researches to have an insight of the distribution of the biomarkers on positive and negative individuals, and on those whose status is unknown. A single main function (`sMSROC`) wraps secondary functions developed to compute the sMS estimator, and the AUC with a confidence interval. This method allows to link the diagnosis and prognosis scenarios via a predictive model which models the relationship between the biomarker and the event under study. The most common probabilistic models (e.g. logistic regression, Cox proportional hazards regression) are implemented out-of-the-box and the user can also enter their own predicted probabilities which can be computed using any other appropriate model. A separate function computes the weighted empirical estimator of the biomarker to get the corresponding estimates for the sensitivity and specificity (second stage). We also implemented three different ways of computing the variance of the AUC and these are available in the package. The package also contains several summarize functions which provide useful numerical and graphical outputs. These include the ROC curve plots, a plot of the evolution of the AUC over time, or the plots of the predictive models.

Appendix

Assume a sample with N subjects. Let $\{\hat{\mathcal{P}}(D|x_1), \cdots, \hat{\mathcal{P}}(D|x_N)\}$ be the individual estimated probabilities of being within the positive group, and let $\{\hat{\sigma}^2(x_1), \cdots, \hat{\sigma}^2(x_N)\}$ be their respective variance. Then, if $\hat{\pi}_N = N^{-1}\sum_{i=1}^{N}\hat{\mathcal{P}}(D|x_i)$, the TVE approximation for the variance of AUC is

$$\text{TVE} = \frac{1}{[\hat{\pi}_N \cdot (1-\hat{\pi}_N])^2} \cdot [\hat{v}_1^2 + \hat{v}_2^2],$$

where

$$\hat{v}_1^2 = N^{-1}\sum_{j=1}^{N}\left\{[\hat{W}_{\text{se}}(x_j) + \hat{W}_{\text{sp}}(x_j)] \cdot \mathcal{P}(D|x_j) - \hat{W}_{\text{sp}}(x_j)\right\}^2, \quad \text{and}$$

$$\hat{v}_2^2 = \left\{N^{-1} \cdot \sum_{j=1}^{N}[\hat{W}_{\text{sp}}(x_j) - \hat{W}_{\text{se}}(x_j)]\ddot{\sigma}_(x_j)\right\}^2,$$

and with

$$\hat{W}_{\text{se}}(x) = N^{-1}\sum_{j=1}^{N}[I_{(x_j,\infty)}(x) - \widehat{Se}(x_j)] \cdot [1 - \hat{\mathcal{P}}(D|x_j)], \quad \text{and}$$

$$\hat{W}_{\text{sp}}(x) = N^{-1}\sum_{j=1}^{N}[I_{(x_j,\infty)}(x) - \widehat{Sp}(x_j)] \cdot \hat{\mathcal{P}}(D|x_j),$$

$I_A(s)$ is the indicator function (takes the value 1 if $s \in A$, and 0 otherwise) and $\widehat{Se}(\cdot)$ and $\widehat{Sp}(\cdot)$ are the estimates for the sensitivity and the specificity, respectively. The most challenging part of approximating the variance is usually the computation of $\hat{\sigma}^2(\cdot)$. When $\hat{\mathcal{P}}(D|x)$ is based on logistic or Cox-type regression models, closed-form equations for estimating the variance are available. However, these equations are based on the Delta-method and the obtained results are sometimes not good estimates. The EVE approximation considers that the proposed AUC estimator variance is similar to the one based on the empirical estimator of the observed subjects, and therefore it could be approximated through

$$\text{EVE} = \frac{N}{N_O} \cdot \left\{\frac{1}{1-\hat{\pi}_N} \cdot < \widehat{\text{Se}}, \widehat{\text{Sp}} > + \frac{1}{\hat{\pi}_N} \cdot < \widehat{\text{Sp}}, \widehat{\text{Se}} >\right\},$$

where given two real functions f and g, $< f, g >= \int f^2 dg - (\int f dg)^2$, and N_O is the number of subjects with complete information (those used for estimating the predictive model).

References

C. Anderson-Bergman. icenReg: Regression models for interval censored data in R. *Journal of Statistical Software*, 81(12):1–23, 2017. URL https://doi.org/10.18637/jss.v081.i12. [p132]

N. N. Basu, S. Ingham, J. Hodson, F. Lalloo, M. Bulman, A. Howell, and D. G. Evans. Risk of contralateral breast cancer in BRCA1 and BRCA2 mutation carriers: a 30-year semi-prospective analysis. *Familial Cancer*, 14(4):531–538, 2015. URL https://doi.org/10.1007/s10689-015-9825-9. [p129]

K. M. Beyene and A. El Ghouch. Time-dependent ROC curve estimation for interval-censored data. *Biometrical Journal*, 64(6):1056–1074, 2022. URL https://doi.org/10.1002/bimj.202000382. [p129, 142]

K. M. Beyene and A. El Ghouch. *cenROC: estimating time-dependent ROC curve and AUC for censored data*, 2023. URL https://CRAN.R-project.org/package=cenROC. R package version 2.0.0. [p129]

K. M. Beyene and A. El Grouch. Smoothed time-dependent receiver operating characteristic curve for right censored survival data. *Statistics in Medicine*, 39(24):3373–3396, 2020. URL https://doi.org/10.1002/sim.8671. [p129]

P. Blanche, J. F. Dartigues, and H. Jacqmin-Gadda. Review and comparison of ROC curve estimators for a time-dependent outcome with marker-dependent censoring. *Biometrical Journal*, 55(5):687–704, 2013a. URL https://doi.org/10.1002/bimj.201200045. [p129]

P. Blanche, J.-F. Dartigues, and H. Jacqmin-Gadda. Estimating and comparing time-dependent areas under receiver operating characteristic curves for censored event times with competing risks. *Statistics in Medicine*, 32(30):5381–5397, 2013b. URL https://doi.org/10.1002/sim.5958. [p129]

L. Chambles and G. Diao. Estimation of time-dependent area under ROC curve for long-term risk prediction. *Statistics in Medicine*, 20(25):3474–3486, 2006. URL https://doi.org/10.1002/sim.2299. [p129, 132]

D. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220, 1972. URL https://doi.org/10.1111/j.2517-6161.1972.tb00899.x. [p136]

S. Díaz-Coto, P. Martínez-Camblor, and N. O. Corral-Blanco. Cumulative/dynamic ROC curve estimation under interval censorship. *Journal of Statistical Computation and Simulation*, 90(9):1570–1590, 2020a. URL https://doi.org/10.1080/00949655.2020.1736071. [p129, 132, 136]

S. Díaz-Coto, P. Martínez-Camblor, and S. Pérez-Fernández. smoothROCtime: an R package for time-dependent ROC curve estimation. *Computational Statistics*, 2020b. URL https://doi.org/10.1007/s00180-020-00955-7. [p129]

S. Díaz-Coto, N. Corral-Blaco, and P. Martínez-Camblor. Two-stage receiver operating-characteristic curve estimator for cohort studies. *The International Journal of Biostatistics*, 17:117–137, 2021. URL https://doi.org/10.1515/ijb-2019-0097. [p130, 131]

J. I. Epstein, L. Egevad, M. B. Amin, B. Delahunt, J. R. Srigley, P. A. Humphrey, and the Grading-Comittee. The 2014 international society of urological pathology (ISUP) consensus conference on gleason grading of prostatic carcinoma. *The American Journal of Surgical Pathology*, 40(2):244–252, 2016. URL https://doi.org/10.1097/PAS.0000000000000530. [p129]

R. Etzioni, M. Pepe, G. Longton, C. Hu, and G. Goodman. Incorporating the time dimension in receiver operating characteristic curves: A case study of prostate cancer. *Medical Decision Making*, 19(3):242–251, 1999. URL https://doi.org/10.1177/0272989X9901900303. [p129]

P. M. Farrell, B. J. Rosenstein, T. B. White, F. J. Accurso, C. Castellani, G. R. Cutting, P. R. Durie, V. A. LeGrys, J. Massie, R. B. Parad, M. J. Rock, P. W. Campbell 3rd, and Cystic-Fibrosis-Foundation. Guidelines for diagnosis of cystic fibrosis in newborns through older adults: Cystic fibrosis foundation consensus report. *The Journal of Pediatrics*, 153(2):S4–S14, 2008. URL https://doi.org/10.1016/j.jpeds.2008.05.005. [p128]

E. Ferreirós, C. Boissonnet, R. Pizarro, P. Merletti, G. Corrado, A. Cagide, and O. Bazzino. Independent prognostic value of elevated C-reactive protein in unstable angina. *Circulation*, 100(19):1958–1963, 1999. URL https://doi.org/10.1161/01.CIR.100.19.1958. [p129]

D. M. Finkelstein. A proportional hazards model for interval-censored failure time data. *Biometrics.*, 42(4):845–854, 1986. URL https://doi.org/10.2307/2530698. [p136]

Y. Foucher, P. Daguin, A. Akl, M. Kessler, M. Ladrière, C. Legendre, H. Kreis, L. Rostaing, N. Kamar, G. Mourad, V. Garrigue, F. Bayle, B. H. de Ligny, M. Büchler, C. Meier, J. P. Daurès, J. P. Soulillou, and M. Giral. A clinical scoring system highly predictive of long-term kidney graft survival. *Kidney International*, 78(12):1288–1294, 2010. URL https://doi.org/10.1038/ki.2010.232. [p139]

D. Gohel and P. Skintzos. *flextable: functions for tabular reporting*, 2023. URL https://CRAN.R-project.org/package=flextable. R package version 0.9.3. [p132]

L. Gonçalves, A. Subtil, M. Rosário Oliveira, and P. De Zea Bermudez. ROC curve estimation: An overview. *Statistical Journal*, 12(1):1–20, 2014. URL https://doi.org/10.57805/revstat.v12i1.141. [p128]

J. Hanley and B. McNeil. The meaning and use of the area under the receiver operating characteristic (ROC) curve. *Radiology*, 20(143):29–36, 1982. URL https://doi.org/10.1148/radiology.143.1.7063747. [p128]

F. E. Harrel. *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordonal Regression and Survival Analysis*. Springer Series in Statistics. Springer International Publishing, 2015. [p136]

F. E. Harrell Jr. *rms: Regression modeling strategies*, 2023. URL https://CRAN.R-project.org/package=rms. R package version 6.7-1. [p132]

P. J. Heagerty and P. Saha-Chaudhuri. *survivalROC: time-dependent ROC curve estimation from censored survival data*, 2022. URL https://CRAN.R-project.org/package=survivalROC. R package version 1.0.3.1. [p129]

P. J. Heagerty, T. Lumley, and M. S. Pepe. Time-Dependent ROC Curves for Censored Survival Data and a Diagnostic Marker. *Biometrics.*, 56(2):337–344, 2000. URL https://doi.org/10.1111/j.0006-341x.2000.00337.x. [p129]

F. Hsieh and B. W. Turnbull. Nonparametric and semiparametric estimation of the receiver operating characteristic curve. *The Annals of Statistics*, 24(1):25–40, 1996. URL https://doi.org/10.1214/aos/1033066197. [p132]

H. Hung and C. Chiang. Optimal composite markers for time-dependent receiver operating characteristic curves with censored survival data. *Scandinavian Journal of Statistics*, 20(37):664–679, 2010. URL https://doi.org/10.1111/j.1467-9469.2009.00683.x. [p129]

C. M. Hurvich, J. S. Simonoff, and C.-L. Tsai. Smoothing parameter selection in nonparametric regression using an improved Akaike information criterion. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 60(2):271–293, 1998. URL https://doi.org/10.1111/1467-9868.00125. [p136]

A. N. Kamarudin, T. Cox, and R. Kolamunnage-Dona. Time-dependent ROC curve analysis in medical research: current methods and applications. *BMC Medical Research Methodology*, 17(53), 2017. URL https://doi.org/10.1186/s12874-017-0332-6. [p129]

L. Li and C. Wu. *tdROC: non-parametric estimation of time-dependent ROC curve for right censored survival data*, 2016. URL https://CRAN.R-project.org/package=tdROC. [p129]

L. Li, T. Greene, and B. Hu. A simple method to estimate the time-dependent receiver operating characteristic curve and the area under the curve with right censored data. *Statistical Methods in Medical Research*, 27(8):2264–2278, 2018. URL https://doi.org/10.1177/0962280216680239. [p129, 132]

J. Lin, Y. Wu, X. Wang, and K. Owzar. *intcensROC: AUC estimation of interval censored survival data*, 2021. URL https://CRAN.R-project.org/package=intcensROC. R package version 0.1.3. [p129]

J. Long, Z. Yang, L. Wang, Y. Han, C. Peng, C. Yan, and D. Yan. Metabolite biomarkers of Type 2 diabetes mellitus and pre-diabetes: a systematic review and meta-analysis. *BMC Endocrine Disorders*, 20(1):SP174, 2020. URL https://doi.org/10.1186/s12902-020-00653-x. [p128]

M. López-Ratón, M. X. Rodríguez-Álvarez, C. Cadarso-Suárez, and F. Gude-Sampedro. OptimalCutpoints: An R package for selecting optimal cutpoints in diagnostic tests. *Journal of Statistical Software*, 61(8):1–36, 2014. URL https://doi.org/10.18637/jss.v061.i08. [p132]

P. Martínez-Camblor. Nonparametric cutoff point estimation for diagnostic decisions with weighted errors. *Revista Colombiana de Estadística*, 34(1):133–146, 2011. URL https://doi.org/10.15446/rce. [p142]

P. Martínez-Camblor and J. C. Pardo-Fernández. Smooth time-dependent receiver operating characteristic curve estimators. *Statistical Methods in Medical Research*, 27(3):651–674, 2018. URL https://doi.org/10.1177/0962280217740786. [p129]

P. Martínez-Camblor, G. F. Bayón, and S. Pérez-Fernández. Cumulative/dynamic ROC curve estimation. *Journal of Statistical Computation and Simulation*, 86(17):3582–3594, 2016. URL https://doi.org/10.1080/00949655.2016.1175442. [p129, 132]

Microsoft and S. Weston. *foreach: Provides foreach looping construct*, 2022. URL https://CRAN.R-project.org/package=foreach. R package version 1.5.2. [p132]

L. Ni and X. H. Wehrens. Cardiac troponin I - more than a biomarker for myocardial ischemia? *Annals of Translational Medicine*, Suppl 1(6):S17, 2018. URL https://doi.org/10.21037/atm.2018.09.07. [p128]

M. S. Pepe. *The Statistical Evaluation of Medical Tests for Classification and Prediction*. Oxford Statistical Sciences Series, 2003. [p128]

S. Pérez-Fernández. *nsROC: non-standard ROC curve analysis*, 2017. URL https://CRAN.R-project.org/package=nsROC. [p128]

S. Pérez-Fernández, P. Martínez-Camblor, P. Filzmoser, and N. Corral. nsROC: An R package for non-standard ROC curve analysis. *The R Journal*, 10(2):55–77, 2018. URL https://doi.org/10.32614/RJ-2018-043. [p128]

S. Potapov, W. Adler, and M. Schmid. *survAUC: estimators of prediction accuracy for time-to-event data*, 2023. URL https://CRAN.R-project.org/package=survAUC. R package version 1.2-0. [p129]

X. Robin, N. Turck, A. Hainard, N. Tiberti, F. Lisacek, J. Sánchez, and M. M¨ uller. pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics*, 12(3):77, 2011. URL https://doi.org/10.1186/1471-2105-12-77. [p128]

M. X. Rodríguez-Álvarez and V. Inácio. ROCnReg: An R package for receiver operating characteristic curve inference with and without covariates. *The R Journal*, 13:525, 2021. URL https://doi.org/10.32614/RJ-2021-066. [p129]

M. C. Sachs. plotROC: a tool for plotting ROC curves. *Journal of Statistical Software*, 79(2):1–19, 2017. URL https://doi.org/10.18637/jss.v079.c02. [p128, 132]

T. Sing, O. Sander, N. Beerenwinkel, and T. Lengauer. ROCR: visualizing classifier performance in R. *Bioinformatics*, 21(20):7881, 2005. URL https://doi.org/10.1093/bioinformatics/bti623. [p128]

X. Song and X. H. Zhou. A semiparametric approach for the covariate-specific ROC curve with survival outcome. *Statistica Sinica*, 18:947–965, 2008. URL http://www.jstor.org/stable/24308524. [p132]

Terry M. Therneau and Patricia M. Grambsch. *Modeling Survival Data: Extending the Cox Model*. Springer, New York, 2000. ISBN 0-387-98784-3. [p132]

B. Turnbull. The empirical distribution function with arbitrarily grouped, censored and truncated data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 38(3):290–295, 1976. URL http://www.jstor.org/stable/2984980. [p136]

H. Uno, T. Cai, L. Tian, and L. J. Wei. Evaluating prediction rules for t-year survivors with censored regression models. *Journal of the American Statistics Association*, 478(102):527–537, 2007. URL https://doi.org/10.1198/016214507000000149. [p129]

J. R. Vidal-Castiñeira, A. López-Vázquez, P. Díaz-Bulnes, S. Díaz-Coto, L. Márquez-Kisinousky, J. Martínez-Borra, C. A. Navascues, P. Sanz-Cameno, A. A. Juan de la Vega, M. Rodríguez, and C. López-Larrea. Genetic contribution of endoplasmic reticulum aminopeptidase 1 polymorphisms to liver fibrosis progression in patients with HCV infection. *Journal of Molecular Medicine*, 98: 1245–1254, 2020. URL https://doi.org/10.1007/s00109-020-01948-1. [p142]

H. Wickham. *ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org. [p132]

J. P. Willems, J. T. Saunders, D. E. Hunt, and J. B. Schorling. Prevalence of coronary heart disease risk factors among rural blacks: A community-based study. *Southern Medical Journal*, 90(8):814–820, 1997. URL https://doi.org/10.1097/00007611-199708000-00008. [p135]

Y. Wu, X. Wang, J. Lin, J. Beilin, and K. Owzar. Predictive accuracy of markers or risk scores for interval censored survival data. *Statistics in Medicine*, 39(18):2437–2446, 2020. URL https://doi.org/10.1002/sim.8547. [p129]

X.-H. Zhou, N. A. Obuchowski, and D. K. McClish. *Statistical Methods in Diagnostic Medicine*. Wiley Blackwell, New York, 2002. [p128]

Susana Díaz-Coto
Department of Orthopaedics, Dartmouth Health, Lebanon, NH, USA
Geisel School of Medicine at Dartmouth, Hanover, NH, USA

Pablo Martínez-Camblor
Faculty of Health Sciences, Universidad Autonoma de Chile, Chile
and
Department of Anesthesiology, Dartmouth Health, Lebanon, NH, USA
Geisel School of Medicine at Dartmouth, Hanover, NH, USA

Norberto Corral-Blanco
Department of Statistics, Operational Research and Mathematics Didactics, University of Oviedo, Oviedo (Asturias), Spain