

survcomp: a package for performance assessment and comparison for survival analysis

Benjamin Haibe-Kains^{1,2,3}, Markus Schröder^{1,5}, John Quackenbush^{1,2},
Christine Desmedt⁴, Christos Sotiriou⁴, and Gianluca Bontempi³

¹*Computational Biology and Functional Genomics Laboratory, Dana-Farber Cancer Institute, Harvard School of Public Health*

²*Center for Cancer Computational Biology, Dana-Farber Cancer Institute*

³*Machine Learning Group, Université Libre de Bruxelles*

⁴*Breast Cancer Translational Research Laboratory, Institut Jules Bordet, Université Libre de Bruxelles*

⁵*Bioinformatics Resource Facility, Center for Biotechnology, Bielefeld University, Germany*

February 4, 2011

Contents

| | | |
|----------|---------------------------------|----------|
| 1 | Introduction | 1 |
| 1.1 | Installation | 2 |
| 1.2 | Further help | 2 |
| 1.3 | Citing | 2 |
| 2 | Quickstart | 3 |
| 2.1 | Overview | 3 |
| 2.1.1 | The Concordance Index | 3 |
| 2.1.2 | The D Index | 4 |
| 2.1.3 | The Hazard Ratio | 5 |
| 2.1.4 | The Brier Score | 6 |
| 3 | Functions in survcomp | 8 |

1 Introduction

The *survcomp* package is providing functions to assess and to compare the performance of risk prediction (survival) models. This package has been primarily used in gene expression studies and the corresponding results have been published in high impact

journals (Bioinformatics, Clinical Cancer Research, Journal of Clinical Oncology, ...). We plan to extend its use to other -omics high-throughput data in a close future.

1.1 Installation

survcomp requires that *survival*, *ipred*, *prodlm*, *survivalROC*, *SuppDists*, *bootstrap* and *R* ($\geq 2.3.0$) are installed. These should be installed automatically when you install *survcomp*. To install *survcomp*, source *biocLite* from bioconductor

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("survcomp")
```

1.2 Further help

To view the *survcomp* description and a summary of all the functions within *survcomp*, type the following:

```
> library(help=survcomp)
```

1.3 Citing

We are delighted if you use this package. Please do email us if you find a bug or have a suggestion. We would be very grateful if you could cite:

B. Haibe-Kains, C. Desmedt, C. Sotiriou and G. Bontempi (2008) A comparative study of survival models for breast cancer prognostication based on microarray data: does a single gene beat them all? *Bioinformatics* **24(19)**:2200-2208.

2 Quickstart

We will very briefly demonstrate some of the functions in *survcomp*.

2.1 Overview

2.1.1 The Concordance Index

The Concordance Index function computes the concordance index for a risk prediction, i.e. the probability that, for a pair of randomly chosen comparable samples, the sample with the higher risk prediction will experience an event before the other sample or belongs to a higher binary class. The usage is:

```
concordance.index(x, surv.time, surv.event, cl, weights, strat, alpha =  
0.05, outx = TRUE, method = c("conservative", "noether", "nam"), na.rm = FALSE)
```

And a small example with random data could look like this:

```
> set.seed(12345)  
> age <- rnorm(100, 50, 10)  
> sex <- sample(0:1, 100, replace = TRUE)  
> stime <- rexp(100)  
> cens <- runif(100, 0.5, 2)  
> sevent <- as.numeric(stime <= cens)  
> stime <- pmin(stime, cens)  
> strat <- sample(1:3, 100, replace = TRUE)  
> cat("survival prediction:\n")
```

survival prediction:

```
> ci <- concordance.index(x = age, surv.time = stime, surv.event = sevent,  
+   strat = strat, method = "noether")  
> ci[1:5]
```

```
$c.index  
[1] 0.5500355
```

```
$se  
[1] 0.03662212
```

```
$lower  
[1] 0.4782575
```

```
$upper
```

```
[1] 0.6218135
```

```
$p.value
```

```
[1] 0.08592802
```

2.1.2 The D Index

The D Index function computes the D index for a risk prediction, i.e. an estimate of the log hazard ratio comparing two equal-sized prognostic groups. This is a natural measure of separation between two independent survival distributions under the proportional hazards assumption. The usage is:

```
D.index(x, surv.time, surv.event, weights, strat, alpha = 0.05, na.rm = FALSE, ...)
```

And a small example with random data could look like this:

```
> set.seed(12345)
> age <- rnorm(100, 50, 10)
> stime <- rexp(100)
> cens <- runif(100, 0.5, 2)
> sevent <- as.numeric(stime <= cens)
> stime <- pmin(stime, cens)
> strat <- sample(1:3, 100, replace = TRUE)
> di <- D.index(x = age, surv.time = stime, surv.event = sevent,
+             strat = strat)
> di[1:8]
```

```
$d.index
```

```
[1] 1.057789
```

```
$coef
```

```
[1] 0.05618072
```

```
$se
```

```
[1] 0.1844491
```

```
$lower
```

```
[1] 0.736878
```

```
$upper
```

```
[1] 1.518456
```

```
$p.value  
[1] 0.7606811
```

```
$n  
[1] 100
```

```
$coxph  
Call:  
coxph(formula = Surv(stime, sevent) ~ strata(sstrat) + z, weights = sweights)
```

```
      coef exp(coef) se(coef)      z      p  
z 0.0562      1.06    0.184 0.305 0.76
```

```
Likelihood ratio test=0.09  on 1 df, p=0.76  n= 100, number of events= 70
```

2.1.3 The Hazard Ratio

The Hazard Ratio function computes the hazard ratio for a risk prediction. The usage is:

```
hazard.ratio(x, surv.time, surv.event, weights, strat, alpha = 0.05, na.rm  
= FALSE, ...)
```

And a small example with random data could look like this:

```
> set.seed(12345)  
> age <- rnorm(100, 50, 10)  
> stime <- rexp(100)  
> cens <- runif(100, 0.5, 2)  
> sevent <- as.numeric(stime <= cens)  
> stime <- pmin(stime, cens)  
> strat <- sample(1:3, 100, replace = TRUE)  
> hr <- hazard.ratio(x = age, surv.time = stime, surv.event = sevent,  
+   strat = strat)  
> hr[1:8]
```

```
$hazard.ratio  
[1] 1.003961
```

```
$coef  
[1] 0.003952968
```

```
$se
```

```

[1] 0.01044472

$lower
[1] 0.9836174

$upper
[1] 1.024725

$p.value
[1] 0.7050847

$n
[1] 100

$coxmodel
Call:
coxph(formula = Surv(stime, sevent) ~ strata(sstrat) + sx, weights = sweights)

```

| | coef | exp(coef) | se(coef) | z | p |
|----|---------|-----------|----------|-------|------|
| sx | 0.00395 | 1.00 | 0.0104 | 0.378 | 0.71 |

Likelihood ratio test=0.14 on 1 df, p=0.705 n= 100, number of events= 70

2.1.4 The Brier Score

The Brier Score function computes all the Brier scores (BSC) and the corresponding integrated Brier score (IBSC) from a risk score, for all the times of event occurrence. The risk score is first transformed in survival probabilities using either a Cox model or the product-limit estimator. The usage is:

```
sbrier.score2proba(data.tr, data.ts, method = c("cox", "prodlim"))
```

And a small example with random data could look like this:

```

> set.seed(12345)
> age <- rnorm(30, 50, 10)
> stime <- rexp(30)
> cens <- runif(30, 0.5, 2)
> sevent <- as.numeric(stime <= cens)
> stime <- pmin(stime, cens)
> dd <- data.frame(time = stime, event = sevent, score = age)
> sb1 <- sbrier.score2proba(data.tr = dd, data.ts = dd, method = "cox")

```

```
> sb2 <- sbrier.score2proba(data.tr = dd, data.ts = dd, method = "prodlm")
> sb1$bsc.integrated

[1] 0.1921649

> sb2$bsc.integrated

[1] 0.1986622
```

3 Functions in survcomp

| FUNCTION | DESCRIPTION |
|--------------------|--|
| D.index | Function to compute the D index |
| sensor.time | Function to artificially censor survival data |
| cindex.comp | Function to compare two concordance indices |
| cindex.comp.meta | Function to compare two concordance indices |
| combine.est | Function to combine estimates |
| combine.test | Function to combine probabilities |
| concordance.index | Function to compute the concordance index for survival or binary class prediction |
| cvpl | Function to compute the CVPL |
| dindex.comp | Function to compare two D indices |
| dindex.comp.meta | Function to compare two concordance indices |
| fisherz | Function to compute Fisher z transformation |
| forestplot.surv | Function to create a Forest Plot |
| getsurv2 | Function to retrieve the survival probabilities at a specific point in time |
| hazard.ratio | Function to estimate the hazard ratio through Cox regression |
| hr.comp | Function to statistically compare two hazard ratios |
| hr.comp.meta | Function to compare two concordance indices |
| hr.comp2 | Function to statistically compare two hazard ratios (alternative interface) |
| iauc.comp | Function to compare two IAUCs through time-dependent ROC curves |
| ibsc.comp | Function to compare two IBSCs |
| km.coxph.plot | Function to plot several Kaplan-Meier survival curves |
| logpl | Function to compute the log partial likelihood of a Cox model |
| no.at.risk | Function to compute the number of individuals at risk |
| sbrier.score2proba | Function to compute the BSCs from a risk score, for all the times of event occurrence |
| score2proba | Function to compute the survival probabilities from a risk score |
| survcomp-package | Performance Assessment and Comparison for Survival Analysis |
| td.sens.spec | Function to compute sensitivity and specificity for a binary classification of survival data |
| tdrocc | Function to compute time-dependent ROC curves |
| test.hetero.est | Function to test the heterogeneity of set of probabilities |
| test.hetero.test | Function to test the heterogeneity of set of probabilities |

References

- [1] Cochran, W. G.: The combination of estimates from different experiments. *Biometrics*, **10**, 101-129. 1954.
- [2] Whitlock, M. C.: Combining probability from independent tests: the weighted Z-method is superior to Fisher's approach. *J. Evol. Biol.*, **18**, 1368-1373. 2005.
- [3] Heagerty, P. J. and Lumley, T. L. and Pepe, M. S.: Time-Dependent ROC Curves for Censored Survival Data and a Diagnostic Marker. *Biometrics*, **56**, 337-344. 2000.
- [4] Efron, B. and Tibshirani, R.: The Bootstrap Method for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science*, **1**, 1-35. 1986.
- [5] Becker, R. A., Chambers, J. M. and Wilks, A. R.: The New S Language. *Wadsworth & Brooks/Cole*, 1988.
- [6] Cox, D. R.: Regression Models and Life Tables. *Journal of the Royal Statistical Society Series B*, **34**, 187-220. 1972.
- [7] Andersen, P. K. and Borgan, O. and Gill, R. D. and Keiding, N.: Statistical Models Based on Counting Processes *Springer*, 1993.
- [8] Brier, G. W.: Verification of forecasts expressed in terms of probabilities. *Monthly Weather Review*, **78**, 1-3. 1950.
- [9] Graf, E. and Schmoor, C. and Sauerbrei, W. and Schumacher, M.: Assessment and comparison of prognostic classification schemes for survival data. *Statistics in Medicine*, **18**, 2529-2545. 1999.
- [10] Wilcoxon, F.: Individual comparisons by ranking methods. *Biometrics Bulletin*, **1**, 80-83. 1945.
- [11] Haibe-Kains, B. and Desmedt, C. and Sotiriou, C. and Bontempi, G.: A comparative study of survival models for breast cancer prognostication based on microarray data: does a single gene beat them all? *Bioinformatics*, **24**:19, 2200-2208. 2008.
- [12] Student: The Probable Error of a Mean. *Biometrika*, **6**, 1-25. 1908.
- [13] R. A. Fisher: Frequency distribution of the values of the correlation coefficient in samples of an indefinitely large population. *Biometrika*, **10**, 507-521. 1915.
- [14] Verweij PJM. and van Houwelingen H: Cross-validation in survival analysis. *Statistics in Medicine*, **12**, 2305-2314. 1993.

- [15] van Houwelingen H, Bruinsma T, Hart AA, van't Veer LJ and Wessels LFA: Cross-validated Cox regression on microarray gene expression data. *Statistics in Medicine*, **25**, 3201-3216. 2006.
- [16] Harrel Jr, F. E. and Lee, K. L. and Mark, D. B.: Tutorial in biostatistics: multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in Medicine*, **15**, 361-387. 1996.
- [17] Pencina, M. J. and D'Agostino, R. B.: Overall C as a measure of discrimination in survival analysis: model specific population value and confidence interval estimation. *Statistics in Medicine*, **23**, 2109-2123. 2004.
- [18] Royston, P. and Sauerbrei, W.: A new measure of prognostic separation in survival data. *Statistics in Medicine*, **23**, 723-748. 2004.