Statistical Methods Supplement: Marker Cut Point Optimization Macro

Evaluating markers of epithelial-mesenchymal transition to identify cancer patients at risk for metastatic disease

Journal: Clinical & Experimental Metastasis

Evan L. Busch, Temitope O. Keku, David B. Richardson, Stephanie M. Cohen, David A. Eberhard, Christy L. Avery, and Robert S. Sandler (University of North Carolina at Chapel Hill; rsandler@med.unc.edu)

1) Introduction

At the end of this supplement we present the SAS code for the macro (“opt”) used to find the cut point for a continuous marker expression variable that yields the best model fit for a Cox regression of the association between the dichotomized marker expression variable and time from surgery to patient death.

For anyone who wishes to use the macro, we describe the features of the code that can be adapted to the investigator’s data and desire to control the output. For ease of reference, every fifth line of code has been numbered. Skipped lines and annotations do not count in the line numberings. The code was written using SAS version 9.3 (SAS Institute, Cary, NC).

2) Features of the code

The macro input is a dataset with one observation per subject and containing the following information: continuous marker expression variables, length of time from surgery to patient death, and any censoring variable to be used in Cox proportional hazards modeling. In our example, the input dataset is called “markercore7” and is read-in in line 3.

For any continuous marker expression variable, the macro orders the expression values of all subjects from least to greatest, identifying the observed range of values. For a given marker expression value, the program dichotomizes the variable at that value, thereby establishing distinct marker-positive and marker-negative groups as defined by the particular cut point.

The dichotomized marker expression variable is fit as the only independent variable in a Cox proportional hazards model of time from surgery to death (PROC PHREG in lines 41-45). In our example, TTE5 is the time-to-event variable, with administrative censoring at 5 years. CENSOR5 is an indicator of whether the subject was censored at 5 years (0=not censored, 1=censored). The Cox model produces a fit statistic for the current iteration of the program.

The macro repeats this process for every continuous expression value in the observed data. The dichotomous marker expression variable is named “cut.” The macro output is a list of expression values and corresponding model fit statistics when marker expression is dichotomized at that particular expression value. The list is ordered from lowest fit statistic (best fit) to highest fit statistic (worst fit). See Section 3 below for an example.

Investigators using the macro may wish to tailor the output at several points. First, the investigator can control what kind of model fit statistics are produced by setting the value of “_n_” in line 49. Setting the value to 1 requests -2 log likelihood statistics, a value of 2 requests

AIC statistics, and 3 requests BIC statistics. We chose to work with BIC statistics and so used a value of 3, as well as naming the fit statistic output variable “bic.”

Second, the macro incorporates a “switch” variable that can be set to 0 or 1. Setting switch=0 produces a complete list of all expression values and corresponding model fit statistics, ordered from lowest to highest fit statistics. Switch=1 restricts the output to the lowest model fit statistic and its corresponding expression value.

The final line of the code calls the macro for a particular continuous marker expression variable: “%opt(EAIWAV_T,0);” (line 85). The items in parentheses are particular values of the general form (continuous marker expression variable, switch variable). Our example expression variable is EAIWAV_T, which stands for “continuous E-cadherin expression measured on the average intensity scale, values assigned as weighted averages of cores, for tumor cores.” We request that the program show the full macro output for this variable and so have set switch=0.

3) Sample macro output

Below is the first 10 rows of output for marker expression variable EAIWAV_T with BIC model fit statistics. Since we had 190 subjects in our dataset, the actual list of output results has as many rows as the number of subjects, minus ties for continuous expression values. These first 10 results are the 10 lowest (best-fitting) BIC values for this expression continuum in our study. E-cadherin was measured on a continuous average intensity scale ranging from 0 to 3.

Result for EAIWAV_T

cut	bic
0.51537	624.441
0.56128	624.976
0.50443	625.234
0.47189	625.410
0.52104	625.965
0.56590	626.022
0.47018	626.048
0.58048	626.208
0.82207	626.332
1.55779	626.346

The top result sets the cut point at an expression value of 0.51537 (what we refer to in the main text as “about 0.52”) and has the best fit of all bivariate associations between dichotomous marker expression status and time-to-death in our observed data. We thus refer to 0.51537 as the statistically-optimal cut point for E-cadherin weighted average. A dichotomous positive/negative status variable was created for E-cadherin weighted average using this cut point. We used this dichotomous variable in the E-cadherin weighted average Cox models presented in Table 3 of the main text.

4) Comparison of optimization macro to receiver operating characteristic (ROC) curves

Our approach is an alternative to ROC curves. Both methods can be used to select a cut point along a continuum of marker expression values based on a criterion that relates marker expression to patient outcomes. The methods differ in two important respects.

First, the form of subject outcomes is different. ROC curves use a binary outcome of whether a subject died (yes/no) while our approach uses continuous time-to-death. It matters whether a patient died 5 months or 50 months after surgery. ROC curves do not account for such distinctions whereas our approach does.

Second, the criterion used to identify a cut point differs between the methods. In ROC curves, the cut point selected is typically the one corresponding to the most upper-left-hand point on a plot of sensitivity versus (1 – specificity) (i.e. true positive rate versus false positive rate). In our approach, the statistically-optimal cut point is the one yielding the best model fit in a bivariate proportional hazards model of dichotomous marker expression and continuous time-to-death.

This difference between cut point selection criteria implies a difference in interpretation between the two methods. Being based on measures of sensitivity and specificity, the cut point selected by an ROC curve is usually interpreted as the one that should be implemented clinically. However, selecting the upper-left-hand corner of the ROC curve implies that false positives and false negatives have clinical consequences of roughly equal importance, which is rarely true. In contrast, the optimal cut point in our approach is a statistical measure of the largest difference in the observed data between hazard functions for marker-positive and marker-negative subjects. Thus, the direct application of our method is to determine whether an association exists between marker expression and time-to-death. The statistically-optimal cut point might or might not be judged to be best for clinical use, but that determination requires consideration of additional information besides model fit alone.

5) SAS code for marker cut point optimization macro

```
%macro opt(var,switch);
/* no missing */
  data _internal_;
    set markercore7;
    if &var^=.;
  run;                                                        /* line 5 */
/* values of &VAR ordered */
  proc freq data=_internal_ noprint;
    tables &var/out=_table_(keep=&var);
  run;
/* number unique values of &VAR to &NN */
  data _null_;
    set _table_ end=end;                                      /* line 10 */
    if end then do;
      nn=put(_n_,8.);
        call symput("nn",nn);
    end;
  run;                                                        /* line 15 */
/* initialize the output data set to _NULL_ */
  data _rsq_;
    set _null_;
  run;
  %do i=1 %to &nn;
/* get the present cut point */
    data _cut_;                                               /* line 20 */
        set _table_;
        rename &var=cut;
        if _n_=&i then do;
          xx=put(&var,25.10);
            call symput("cut",xx);                            /* line 25 */
            output;
        end;
      run;
/* merge and assign to groups */
      data _use_;
        if _n_=1 then set _cut_;                              /* line 30 */
        set _internal_;
        group=(&var<=cut);
        x1=group*&var;
        x0=(1-group)*&var;
      run;                                                    /* line 35 */
      proc datasets;
        delete _cut_;
      run;
      quit;
/* model */
      ods listing close;                                      /* line 40 */
      proc phreg data=_use_;
        model TTE5*CENSOR5(1)=x1;
        ods output fitstatistics=_fit_(keep=withcovariates);
      run;
      quit;                                                   /* line 45 */
      ods listing;
    data _fit_;
      set _fit_;
      if _n_=3;
      rename withcovariates=rsquare;                          /* line 50 */
    run;
      data _rsq_;
        set _rsq_ _fit_(in=in);
        keep cut bic;
        if in then do;                                        /* line 55 */
          cut=&cut;
            bic=rsquare;
        end;
      run;
      proc datasets;                                          /* line 60 */
        delete _use_ _fit_;
      run;
      quit;
  %end;
  proc datasets;                                              /* line 65 */
    delete _table_ _internal_;
  run;
  quit;
  proc sort data=_rsq_ out=_rsq_;
    by bic;                                                   /* line 70 */
  run;
  title "Result for %upcase(&var)";
  proc print noobs data=_rsq_
  %if &switch=1 %then %do;
    (obs=1)                                                   /* line 75 */
  %end;
  ;
  run;
  title;
  proc datasets;                                              /* line 80 */
    delete _rsq_;
  run;
  quit;
%mend;

/* call macro for each continuous expression variable */

%opt(EAIWAV_T,0);                                             /* line 85 */
```