Supplementary Table S1 SAS codes for the propensity score matching

```
/***1. SAS coding before applying macro***/
proc logistic
data=h.Cohort_high;
class Tepisode(ref='1') sex(ref='1') age_g(ref='1') PUD_svr(ref='0') PUD_nonsvr(ref='0')
cirrhosis(ref='0') DM(ref='0') CKD(ref='0') HemorrS(ref='0') cancer(ref='0') alcoholism(ref='0')
GERD(ref='0') IBD(ref='0') tpenia(ref='0') HF(ref='0') HTN(ref='0') ACO(ref='0') APLT(ref='0')
steroid(ref='0') SSRI(ref='0') LASA(ref='0') PYI(ref='0') RFcount(ref='1') CCI_g(ref='0'); /*insert
reference value for categorical variable*/
model GPA = sex age_g PUD_svr PUD_nonsvr cirrhosis HemorrS CCI_g; /*insert matching
variables*/
output out=h.matching_high pred=prob_3 xbETA=logit;
run;

/*conpute standard deviation of the logit of the propensity score */
proc means std data=h.matching_high;
var logit;
output out=stddata (keep=std) std=std;
run;
/*Calipers of width 0.2 standard deviations of the logit of PS*/
data stddata;
set stddata;
std=0.2*std;
run;
/*creat macro variable that contains the width of the caliper for matching*/
data _null_;
set stddata;
call symput ('stdcal', std);
run;
/*Match subjects on the logit of the propensity score.*/
data h.matching_high2;
set h.matching_high;
trtm=GPA;
run;

proc sort
data=h.matching_high2;
by trtm;
run;

data h.matching_high3;
set h.matching_high2;
no=_n_;
run;

data case cont;
set   h.matching_high3;
if trtm=1 then output case;
if trtm=0 then output cont;
run;

/*Apply SAS macro file for greedy matching*/
  %include '/library/match.sas';


/***2. SAS macro file for greedy matching (h.match.SAS) using 2004 Mayo Clinic College of Medicine
program***/
```

```
/*-------------------------------------------------------------------*
  | MACRO NAME    : match
  | SHORT DESC    : Match one or more controls to each of a set
  |                 of cases (rendered somewhat obsolete by the
  |                 newer macros %vmatch and %gmatch)
  *-------------------------------------------------------------------*
  | CREATED BY    : Bergstralh, Erik                  (04/07/2004 16:15)
  |               : Kosanke, Jon
  *-------------------------------------------------------------------*
  | PURPOSE
  |
  | ***NOTE - This macro has been replaced by %gmatch and %vmatch,
  | depending on whether one wants greedy matching or optimal matching.
  |
  | Macro name: %match
  |
  | Authors: Jon Kosanke and Erik Bergstralh
  |
  | Date: April 25, 1995
  |
  | Macro function:
  |
  | The purpose of this macro is to match 1 or more controls(from a total
  | of M) for each of N cases.    The controls may be matched to the cases by
  | one or more factors(X's).    The control selected for a particular
  | case(i) will be the control(j) closest to the case in terms of Dij.
  | Dij is just the weighted sum of the absolute differences between the
  | case and control matching factors.    I.e.,
  |
  |        Dij= SUM { W.k*ABS(X.ik-X.jk) }, where the sum is over the number
  |                                          of matching factors X(with index
  |                                          k) and W.k = the weight assigned
  |                                          to matching factor k and X.ik =
  |                                          the value of variable X(k) for
  |                                          subject i.
  |
  | The control(j) selected for a case(i) is that with the smallest Dij
  | which is less than or equal DMAX(and which is compatible with the DMAXK
  | option below), where DMAX is defined by the user.    In the case of ties,
  | the first one encountered will be used.          The higher the user-defined
  | weight, the more likely it is that the case and control will be matched
  | on the factor.    Assign large weights (relative to the other weights) to
  | obtain exact matches for two-level factors such as gender.
  |
  | Using the GREEDY method, once a match is made it is never broken.    This
  | may result in inefficiencies if a previously matched control would be a
  | better match for the current case than those controls currently
  | available.
  |
  | The OPTIMAL method uses PROC NETFLOW from SAS/OR to find the set of
  | matches that minimizes the sum of Dij over all possible sets of
  | matches.    The OPTIMAL method also has an option for a variable number
  | of controls per case.
  |
  | The macro checks for missing values of matching variables and the time
  | variable(if specified) and deletes those observations from the case
  | and control datasets.
  |
```

```
| Call statement:
| %match(case=,control=,idca=,idco=,
|          mvars=,wts=,dmaxk=,dmax=,
|          time=,
|          method=,
|          ncontls=,seedca=,seedco=,
|          mincont=,maxcont=,maxiter=,
|          out=,outnmca=,outnmco=,print=);
|
| Parameter definitions(R=required parameter):
|
|   R       case=SAS data set of cases.   Must contain the IDCA variable and
|                the matching variables.
|
|   R       control=SAS data set of possible controls.   Must contain the
|                IDCO variable and the matching variables.   Note the macro
|                assumes that the cases and controls are in different
|                data sets.
|            FOR RISK SET MATCHING THIS DATA SET SHOULD INCLUDE BOTH CASES
|            AND CONTROLS.
|
|   R       idca=ID variable for the cases.
|
|   R       idco=ID variable for the controls.
|
|          time=time variable used to define risk sets.   Matches are only
|                valid if the control time > case time.
|
|   R       mvars=list of numeric matching variables common to both case and
|                control data sets.   For example, mvars=male age birthyr.
|
|   R       wts=list of non-negative weights corresponding to each matching
|               variable.   For example wts=10 2 1 corresponding to male, age
|               and birthyr as in the above example.
|
|          dmaxk=list of non-negative values corresponding to each matching
|                variable.   These numbers are the largest possible absolute
|                differences compatible with a valid match.   Cases will
|                NOT be matched to a control if ANY of the INDIVIDUAL
|                matching factor   differences are >DMAXK.   This optional
|                parameter allows one to form matches of the type male+/-0,
|                age+/-2, birth year+/-5 by specifying DMAXK=0 2 5.
|                Given that a possible control meets the DMAXK criteria,
|                the macro selects the control with the smallest Dij.
|                If this list is shorter that the MVARS list, 1-1 matching
|                will be done until the DMAXK list is exhausted.   If
|                this list is longer, the extra DMAXK values are ignored.
|
|
|          dmax=largest value of Dij considered to be a valid match.   If
|                you want to match exactly on a two-level factor(such as
|                gender coded as 0 or 1) then assign DMAX to be less than
|                the weight for the factor.   In the example above, one could
|                use wt=10 for male and dmax=9.   Leave DMAX blank if any
|                Dij is a valid match.   One would typically NOT use both
|                DMAXK and DMAX.   The only advantage to using both, would be
|                to further restrict potential matches that meet the
|                DMAXK criteria.
|
```

```
|   R        method= GREEDY or OPTIMAL.    See reference below.
|
|            ncontls=fixed number of controls to match to each case.    The
|                    default is 1.    Using the GREEDY method with multiple
|                    controls per case, the algorithm will first match every
|                    case to one control and then again match each case to a
|                    second control, etc.    Controls selected on the first
|                    pass will be stronger matches than those selected in
|                    later rounds.    The output data set contains a variable
|                    (cont_n) which indicates on which round the control was
|                    selected.    This option is ignored if a variable number
|                    of controls is to be used with the OPTIMAL method(see
|                    MINCONT and MAXCONT parameters below).
|
| **** Options specific to GREEDY method ******************************
|   R        seedca=seed value used to randomly sort the cases prior to
|                   matching using the GREEDY method.    This positive integer
|                   must be less than (2**31)-1 and will be used as input to
|                   the RANUNI function.    The greedy matching algorithm is
|                   order dependent which, among other things means that
|                   cases matched first will be on average more similar to
|                   their controls than those matched last(as the number of
|                   control choices will be limited).    If the matching order
|                   is related to confounding factors (possibly age or
|                   calendar time) then biases may result.    Therefore it is
|                   generally considered good practice when using the GREEDY
|                   method to randomly sort both the cases and controls
|                   before beginning the matching process.
|
|    R       seedco=seed value used to randomly sort the controls prior to
|                   matching using the GREEDY method.    This seed value must
|                   also be an integer less than (2**31)-1.
|
| **** Options specific to OPTIMAL method ***************************
|            mincont=minimum number of controls per case using the OPTIMAL
|                    method with a variable number of controls(see Section
|                    3.3 of Rosenbaum).    MINCONT must be >=1.
|
|            maxcont=maximum number of controls per case using the OPTIMAL
|                    method with a variable number of controls(see Section
|                    3.3 of Rosenbaum).
|                    MAXCONT must be >= MINCONT and <= M-N+1.
|
|            maxiter=maximum number of iterations for PROC NETFLOW to use
|                    under the OPTIMAL method.    Default value is 100000.
|
| **** OUTPUT options applicable to either method ******************
| print= Option to print data for matched cases. Use PRINT=y to
|            print data and PRINT=n or blank to not print.    Default is y.
|
|            out=name of SAS data set containing the results of the matching
|                 process.    Unmatched cases are not included.    See outnm
|                 below.    The default name is __out.    This data set will have
|                 the following layout:
|
|              Case_id   Cont_id   Cont_n   Dij   Delta_caco MVARS_ca   MVARS_co
|                 1         67        1      5.2   (Differences & actual
|                 1         78        2      6.1    values for matching factors
|                 2         52        1      2.9    for cases & controls)
```

```
|                  2          92         2        3.1
|                  .          .          .        .
|                  .          .          .        .
|
|          outnmca=name of SAS data set containing NON-matched cases.
|                    Default name is __nmca .
|
|          outnmco=name of SAS data set containing NON-matched controls.
|                    Default name is __nmco .
|
|
|   References:   Bergstralh, EJ and Kosanke JL(1995).   Computerized
|                  matching of controls.   Section of Biostatistics
|                  Technical Report 56.   Mayo Foundation.
|
|                  Paul R.   Rosenbaum.   Optimal matching for observational
|                  studies.   JASA, 84(408), pp. 1024-1032, 1989.
|
|   Example: 1-1 matching by male(exact), age(+-2) and year(+-5).
|              The wt for male is not relevant, as only exact matches
|              on male will be considered.   The weight for age(2) is
|              double that for year(1).
|
|        A. Optimal method.
|
|         %match(case=case,control=cont,idca=clinic,idco=clinic,
|                  mvars=male age_od yr_od,maxiter=10000,
|                  wts=2 2 1, dmaxk=0 2 5,out=mtch,
|                  method=optimal);
|
|        B. Greedy method.
|
|         %match(case=case,control=cont,idca=clinic,idco=clinic,
|                  mvars=male age_od yr_od,
|                  wts=2 2 1, dmaxk=0 2 5,out=mtch,
|                  method=greedy,seedca=87877,seedco=987973);
|
*-----------------------------------------------------------------*
| OPERATING SYSTEM COMPATIBILITY
|
| UNIX SAS v8     :     YES
| UNIX SAS v9     :
| MVS SAS v8      :
| MVS SAS v9      :
| PC SAS v8       :
| PC SAS v9       :
*-----------------------------------------------------------------*
| Copyright 2004 Mayo Clinic College of Medicine.
|
| This program is free software; you can redistribute it and/or
| modify it under the terms of the GNU General Public License as
| published by the Free Software Foundation; either version 2 of
| the License, or (at your option) any later version.
|
| This program is distributed in the hope that it will be useful,
| but WITHOUT ANY WARRANTY; without even the implied warranty of
| MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
| General Public License for more details.
*-----------------------------------------------------------------*/
```

```
%MACRO MATCH(CASE=,CONTROL=,IDCA=,IDCO=,MVARS=,WTS=,DMAXK=,DMAX=,
             NCONTLS=1, TIME=,
             METHOD=,SEEDCA=,SEEDCO=,MAXITER=100000,PRINT=y,
             OUT=__OUT,OUTNMCA=__NMCA,OUTNMCO=__NMCO,MINCONT=,MAXCONT=);

  %LET BAD=0;
  %IF %LENGTH(&CASE)=0 %THEN %DO;
    %PUT ERROR: NO CASE DATASET SUPPLIED;
    %LET BAD=1;
  %END;
  %IF %LENGTH(&CONTROL)=0 %THEN %DO;
    %PUT ERROR: NO CONTROL DATASET SUPPLIED;
    %LET BAD=1;
  %END;
  %IF %LENGTH(&IDCA)=0 %THEN %DO;
    %PUT ERROR: NO IDCA VARIABLE SUPPLIED;
    %LET BAD=1;
  %END;
  %IF %LENGTH(&IDCO)=0 %THEN %DO;
    %PUT ERROR: NO IDCO VARIABLE SUPPLIED;
    %LET BAD=1;
  %END;
  %IF %LENGTH(&MVARS)=0 %THEN %DO;
    %PUT ERROR: NO MATCHING VARIABLES SUPPLIED;
    %LET BAD=1;
  %END;
  %IF %LENGTH(&WTS)=0 %THEN %DO;
    %PUT ERROR: NO WEIGHTS SUPPLIED;
    %LET BAD=1;
  %END;
  %IF %UPCASE(&METHOD)=GREEDY %THEN %DO;
    %IF %LENGTH(&SEEDCA)=0 %THEN %DO;
      %PUT ERROR: NO SEEDCA VALUE SUPPLIED;
      %LET BAD=1;
    %END;
    %IF %LENGTH(&SEEDCO)=0 %THEN %DO;
      %PUT ERROR: NO SEEDCO VALUE SUPPLIED;
      %LET BAD=1;
    %END;
  %END;
  %IF %LENGTH(&OUT)=0 %THEN %DO;
    %PUT ERROR: NO OUTPUT DATASET SUPPLIED;
    %LET BAD=1;
  %END;
  %IF %UPCASE(&METHOD)^=GREEDY & %UPCASE(&METHOD)^=OPTIMAL %THEN %DO;
    %PUT ERROR: METHOD MUST BE GREEDY OR OPTIMAL;
    %LET BAD=1;
  %END;
  %IF (&MINCONT=    AND &MAXCONT^= ) OR (&MINCONT^=    AND &MAXCONT= )
  %THEN %DO;
    %PUT ERROR: MINCONT AND MAXCONT MUST BOTH BE SPECIFIED;
    %LET BAD=1;
  %END;
  %LET NVAR=0;
  %DO %UNTIL(%SCAN(&MVARS,&NVAR+1,' ')= );
    %LET NVAR=%EVAL(&NVAR+1);
  %END;
  %LET NWTS=0;
```

```
%DO %UNTIL(%QSCAN(&WTS,&NWTS+1,' ')= );
    %LET NWTS=%EVAL(&NWTS+1);
%END;
%IF &NVAR^= &NWTS %THEN %DO;
    %PUT ERROR: #VARS MUST EQUAL #WTS;
    %LET BAD=1;
%END;
%LET NK=0;
%IF %QUOTE(&DMAXK)^=    %THEN %DO %UNTIL(%QSCAN(&DMAXK,&NK+1,' ')= );
    %LET NK=%EVAL(&NK+1);
%END;
%IF &NK>&NVAR %THEN %LET NK=&NVAR;
%DO I=1 %TO &NVAR;
    %LET V&I=%SCAN(&MVARS,&I,' ');
%END;
%IF &NWTS>0 %THEN %DO;
       DATA _NULL_;
       %DO I=1 %TO &NWTS;
            %LET W&I=%SCAN(&WTS,&I,' ');
            IF &&W&I<0 THEN DO;
                 PUT 'ERROR: WEIGHTS MUST BE NON-NEGATIVE';
                 CALL SYMPUT('BAD','1');
            END;
       %END;
       RUN;
%END;
%IF &NK>0 %THEN %DO;
       DATA _NULL_;
       %DO I=1 %TO &NK;
            %LET K&I=%SCAN(&DMAXK,&I,' ');
            IF &&K&I<0 THEN DO;
                 PUT 'ERROR: DMAXK VALUES MUST BE NON-NEGATIVE';
                 CALL SYMPUT('BAD','1');
            END;
       %END;
       RUN;
%END;
%MACRO DIJ;
    %DO I=1 %TO &NVAR-1;
        &&W&I*ABS(__CA&I-__CO&I) +
    %END;
    &&W&NVAR*ABS(__CA&NVAR-__CO&NVAR);
%MEND DIJ;
%MACRO MAX1;
    %IF &DMAX^= %THEN %DO;
        & __D<=&DMAX
    %END;
    %DO I=1 %TO &NK;
        & ABS(__CA&I-__CO&I)<=&&K&I
    %END;
%MEND MAX1;
%MACRO MAX2;
    %IF &DMAX= & &NK=0 %THEN %DO;
        %IF &time^= %then %do;
            if __cotime>__catime then
        %end;
        output;
    %end;
    %IF &DMAX^= & &NK=0 %THEN %DO;
```

```
        IF _COST_<=&DMAX
        %if &time^= %then %do;
            & __cotime>__catime
        %end;
        THEN OUTPUT;
    %END;
    %IF &DMAX= & &NK>0 %THEN %DO;
        IF ABS(__CA1-__CO1)<=&K1
        %DO I=2 %TO &NK;
            & ABS(__CA&I-__CO&I)<=&&K&I
        %END;
        %if &time^= %then %do;
            & __cotime>__catime
        %end;
        THEN OUTPUT;
    %END;
    %IF &DMAX^= & &NK>0 %THEN %DO;
        IF _COST_<=&DMAX
        %DO I=1 %TO &NK;
            & ABS(__CA&I-__CO&I)<=&&K&I
        %END;
        %if &time^= %then %do;
            & __cotime>__catime
        %end;
        THEN OUTPUT;
    %END;
%MEND MAX2;
%MACRO LBLS;
    %DO I=1 %TO &NVAR;
        __CA&I="&&V&I/CASE"
        __CO&I="&&V&I/CONTROL"
        __DIF&I="&&V&I/ABS. DIFF "
        __WT&I="&&V&I/WEIGHT"
    %END;
%MEND LBLS;
%MACRO VBLES;
    %DO I=1 %TO &NVAR;
        __DIF&I
    %END;
    %DO I=1 %TO &NVAR;
        __CA&I __CO&I
    %END;
%MEND VBLES;
%MACRO GREEDY;
 %GLOBAL BAD2;
    DATA __CASE; SET &CASE;
          %DO I=1 %TO &NVAR;
                %LET MISSTEST=%SCAN(&MVARS,&I,' ');
                IF &MISSTEST=. THEN DELETE;
          %END;
          %IF &TIME^= %THEN %DO;
                IF &TIME=. THEN DELETE;
          %END;
    DATA __CASE; SET __CASE END=EOF;
     KEEP __IDCA __CA1-__CA&NVAR __R &mvars
     %if &time^= %then %do;
            __catime
        %end;
        ;
```

```
    __IDCA=&IDCA;
    %if &time^= %then %do;
        __catime=&time;
    %end;
    %DO I=1 %TO &NVAR;
        __CA&I=&&V&I;
    %END;
    SEED=&SEEDCA;
    __R=RANUNI( SEED    );
    IF EOF THEN CALL SYMPUT('NCA',_N_);
PROC SORT; BY __R __IDCA;
DATA __CONT; SET &CONTROL;
       %DO I=1 %TO &NVAR;
            %LET MISSTEST=%SCAN(&MVARS,&I,' ');
            IF &MISSTEST=. THEN DELETE;
       %END;
       %IF &TIME^= %THEN %DO;
            IF &TIME=. THEN DELETE;
       %END;
DATA __CONT; SET __CONT END=EOF;
 KEEP __IDCO __CO1-__CO&NVAR __R &mvars
  %if &time^= %then %do;
       __cotime
  %end;
  ;
    __IDCO=&IDCO;
    %if &time^= %then %do;
        __cotime=&time;
    %end;
    %DO I=1 %TO &NVAR;
        __CO&I=&&V&I;
    %END;
    SEED=&SEEDCO;
    __R=RANUNI( SEED    );
    IF EOF THEN CALL SYMPUT('NCO',_N_);
RUN;
%LET BAD2=0;
%IF &NCO < %EVAL(&NCA*&NCONTLS) %THEN %DO;
    %PUT ERROR: NOT ENOUGH CONTROLS TO MAKE REQUESTED MATCHES;
    %LET BAD2=1;
%END;
%IF &BAD2=0 %THEN %DO;
    PROC SORT; BY __R __IDCO;
    DATA __MATCH;
     KEEP __IDCA __CA1-__CA&NVAR __DIJ __MATCH __CONT_N
     %if &time^= %then %do;
         __catime __cotime
     %end;
     ;
     ARRAY __USED(&NCO) $ 1 _TEMPORARY_;
       DO __I=1 TO &NCO;
           __USED(__I)='0';
       END;
       DO __I=1 TO &NCONTLS;
           DO __J=1 TO &NCA;
               SET __CASE POINT=__J;
               __SMALL=.;
               __MATCH=.;
               DO __K=1 TO &NCO;
```

```
                    IF __USED(__K)='0' THEN DO;
                        SET __CONT POINT=__K;
                        __D=%DIJ
                        IF __d^=. & (__SMALL=. | __D<__SMALL) %MAX1
                        %if &time^= %then %do;
                            & __cotime > __catime
                        %end;
                        THEN DO;
                            __SMALL=__D;
                            __MATCH=__K;
                            __DIJ=__D;
                            __CONT_N=__I;
                        END;
                    END;
                END;
                IF __MATCH^=. THEN DO;
                    __USED(__MATCH)='1';
                    OUTPUT;
                END;
            END;
        END;
        STOP;
    DATA &OUT;
     SET __MATCH;
     SET __CONT POINT=__MATCH;
     KEEP __IDCA __IDCO __CONT_N __DIJ __CA1-__CA&NVAR
          __CO1-__CO&NVAR __DIF1-__DIF&NVAR __WT1-__WT&NVAR
          %if &time^= %then %do;
              __catime __cotime
          %end;
    ;
     LABEL __IDCA="&IDCA/CASE"
           __IDCO="&IDCO/CONTROL"
           %if &time^= %then %do;
               __catime="&time/CASE"
               __cotime="&time/CONTROL"
           %end;
           __CONT_N='CONTROL/NUMBER'
           __DIJ='DISTANCE/D_IJ'
           %LBLS;
        %DO I=1 %TO &NVAR;
           __DIF&I=abs(__CA&I-__CO&I);
           __WT&I=&&W&I;
        %END;
  %END;
%MEND GREEDY;
%MACRO OPTIMAL;
 %GLOBAL BAD2;
  DATA __CASE; SET &CASE;
      %DO I=1 %TO &NVAR;
          %LET MISSTEST=%SCAN(&MVARS,&I,' ');
          IF &MISSTEST=. THEN DELETE;
      %END;
      %IF &TIME^= %THEN %DO;
          IF &TIME=. THEN DELETE;
      %END;
  DATA __CASE; SET __CASE END=EOF;
   KEEP __IDCA __CA1-__CA&NVAR &mvars
     %if &time^= %then %do;
```

```
        __catime
    %end;
    ;
    __IDCA=&IDCA;
    %if &time^= %then %do;
        __catime=&time;
    %end;
    %DO I=1 %TO &NVAR;
        __CA&I=&&V&I;
    %END;
    IF EOF THEN CALL SYMPUT('NCA',_N_);
DATA __CONT; SET &CONTROL;
      %DO I=1 %TO &NVAR;
            %LET MISSTEST=%SCAN(&MVARS,&I,' ');
            IF &MISSTEST=. THEN DELETE;
      %END;
      %IF &TIME^= %THEN %DO;
            IF &TIME=. THEN DELETE;
      %END;
DATA __CONT; SET __CONT END=EOF;
 KEEP __IDCO __CO1-__CO&NVAR &mvars
    %if &time^= %then %do;
        __cotime
    %end;
    ;
    __IDCO=&IDCO;
    %if &time^= %then %do;
        __cotime=&time;
    %end;
    %DO I=1 %TO &NVAR;
        __CO&I=&&V&I;
    %END;
    IF EOF THEN CALL SYMPUT('NCO',_N_);
RUN;
%LET BAD2=0;
%IF &NCO < %EVAL(&NCA*&NCONTLS) %THEN %DO;
    %PUT ERROR: NOT ENOUGH CONTROLS TO MAKE REQUESTED MATCHES;
    %LET BAD2=1;
%END;
%IF &BAD2=0 %THEN %DO;
    DATA __DIST1;
     SET __CASE;
     LENGTH __FROM __TO $ 80;
       DO I=1 TO &NCO;
          SET __CONT POINT=I;
          _COST_=%DIJ;
          __FROM=left(__IDCA);
          __TO=left(trim(__IDCO) || '_co');
          _CAPAC_=1;
          IF _COST_^=. THEN DO;
              %MAX2
          END;
       END;
       DATA __GOODCO;
        SET __DIST1;
        KEEP __IDCO;
       PROC SORT; BY __IDCO;
       DATA __GOODCO;
        SET __GOODCO; BY __IDCO;
```

```
        IF FIRST.__IDCO;
    data _null_;
        i=1;
        set __goodco point=i nobs=n;
        call symput('newcont',n);
        stop;
    DATA __DIST2;
     LENGTH __FROM __TO $ 80;
        DO I=1 TO N;
            SET __GOODCO POINT=I NOBS=N;
            __FROM=left(trim(__IDCO) || '_co');
            __TO='SK';
            _COST_=0;
            _CAPAC_=1;
            OUTPUT;
        END;
    STOP;
    DATA __GOODCA;
     SET __DIST1;
     KEEP __IDCA;
    PROC SORT; BY __IDCA;
    DATA __GOODCA;
     SET __GOODCA; BY __IDCA;
        IF FIRST.__IDCA;
    DATA __DIST3;
     LENGTH __FROM __TO $ 80;
        DO I=1 TO N;
            SET __GOODCA POINT=I NOBS=N;
            __FROM='SC';
            __TO=left(__idca);
            _COST_=0;
            %if &mincont= %then %do;
                _CAPAC_=&NCONTLS;
            %end;
            %else %do;
                _capac_=&mincont;
            %end;
            OUTPUT;
        END;
        %if &mincont^= %then %do;
            __from='SC';
            __to='EXTRA';
            _capac_=&newcont-&mincont*n;
            _cost_=0;
            output;
            do i=1 to n;
                set __goodca point=i;
                __from='EXTRA';
                __to=left(__idca);
                _cost_=0;
                _capac_=&maxcont-&mincont;
                output;
            end;
        %end;
        CALL SYMPUT('NEWCASE',N);
    STOP;
    DATA __DIST;
     SET __DIST1 __DIST2 __DIST3;
%LET DEM=%EVAL(&NEWCASE*&NCONTLS);
```

```
        PROC NETFLOW
            MAXIT1=&MAXITER
            %if &mincont= %then %do;
                DEMAND=&DEM
            %end;
            %else %do;
                demand=&newcont
            %end;
            SOURCENODE='SC'
            SINKNODE='SK'
            ARCDATA=__DIST
            ARCOUT=__MATCH;
         TAIL __FROM;
         HEAD __TO;
        DATA __OUT;
         SET __MATCH;
            IF _FLOW_>0 & __FROM^in ('SC' 'EXTRA') & __TO^='SK';
            __DIJ=_FCOST_;
            %DO I=1 %TO &NVAR;
                __DIF&I=abs(__CA&I-__CO&I);
                __WT&I=&&W&I;
            %END;
        PROC SORT; BY __IDCA __DIJ;
        DATA &OUT;
         SET __OUT; BY __IDCA;
            drop __from -- _status_;
            IF FIRST.__IDCA THEN __CONT_N=0;
            __CONT_N+1;
         LABEL __IDCA="&IDCA/CASE"
                __IDCO="&IDCO/CONTROL"
                %if &time^= %then %do;
                    __catime="&time/CASE"
                    __cotime="&time/CONTROL"
                %end;
                __CONT_N='CONTROL/NUMBER'
                __DIJ='DISTANCE/D_IJ'
                %LBLS;
    %END;
%MEND OPTIMAL;
%IF &BAD=0 %THEN %DO;
    %IF %UPCASE(&METHOD)=GREEDY %THEN %DO;
        %GREEDY
    %END;
    %ELSE %DO;
        %OPTIMAL
    %END;
    %IF &BAD2=0 %THEN %DO;
        PROC SORT DATA=&OUT; BY __IDCA __CONT_N;
        proc sort data=__case; by __IDCA;
        data &outnmca; merge __case
                &out(in=__inout where=(__cont_n=1)); by __idca;
                if __inout=0; **non-matches;

        proc sort data=__cont; by __IDCO;
        proc sort data=&out; by __IDCO;
        data &outnmco; merge __cont
                &out(in=__inout); by __idco;
                if __inout=0; **non-matched controls;
        proc sort data=&out; by __IDCA; **re-sort by case id;
```

```
            %if %upcase(&print)=Y %then %do;
            PROC PRINT data=&out LABEL SPLIT='/';
             VAR __IDCA __IDCO __CONT_N
             %if &time^= %then %do;
                 __catime __cotime
             %end;
             __DIJ %VBLES;
             sum __dij;
            title9'Data listing for matched cases and controls';
            footnote
      "match macro: case=&case control=&control idca=&idca idco=&idco";
            footnote2
"     mvars=&mvars    wts=&wts dmaxk=&dmaxk dmax=&dmax ncontls=&ncontls";
            %if &time^= %then %do;
   footnote3"time=&time    method=&method    seedca=&seedca    seedco=&seedco";
            %end;
            %else %do;
              footnote3"     method=&method    seedca=&seedca    seedco=&seedco";
            %end;
            footnote4"     out=&out     outnmca=&outnmca    outnmco=&outnmco";
            run;
            title9'Summary data for matched cases and controls';
            proc means data=&out n mean sum min max; class __cont_n;
             var __dij
              %if &nvar >=2 %then %do; __dif1-__dif&nvar    __ca1-__ca&nvar
                                  %if &time^= %then %do;
                                      __catime
                                  %end;
                                  __co1-__co&nvar
                                  %if &time^= %then %do;
                                      __cotime
                                  %end;
                                  ;
              %end;
              %else %do;
                                  __dif1 __ca1
                                  %if &time^= %then %do;
                                      __catime
                                  %end;
                                  __co1
                                  %if &time^= %then %do;
                                      __cotime
                                  %end;
                                  ;
              %end;
            run;
            proc means data=&outnmca n mean sum min max; var &mvars;
            title9'Summary data for NON-matched cases';
            run;
            proc means data=&outnmco n mean sum min max; var &mvars;
            title9'Summary data for NON-matched controls';
            run;
            %end;
         %END;
     %END;
      title9; footnote;
      run;
%MEND MATCH;
```

```
/***3. SAS coding after applying macro***/

%match(case=case, control=cont, idca=no, idco=no, mvars=logit, wts=1, dmaxk=&stdcal, out=mtch,
ncontls=2,
method=greedy, seedca=87877, seedco=987973); /*insert number of controls to match to each case
at 'ncontls='*/

data mtch;
set mtch;
match_id= __IDCA;
run;

data m_case;
set mtch;
keep match_id __IDCA;
rename __IDCA=no; run;

data m_control;
set mtch;
keep match_id __IDCO;
rename   __IDCO=no; run;

proc sort
data=m_case nodupkey;
by no; run;

proc sort
data=case;
by no; run;

data m_case;
merge m_case (in=in1) case;
by no;
if in1; run;

proc sort
data=m_control nodupkey;
by no; run;

proc sort
data=cont;
by no; run;

data m_control;
merge m_control (in=in1) cont;
by no;
if in1; run;

data h.matching_high4;
set m_case m_control;
run;

proc freq
data=h.matching_high4;
tables GPA;
run;
```

```
proc freq data=h.matching_high4;
tables GPA*(sex age_g PUD_svr PUD_nonsvr cirrhosis HemorrS CCI_g) /chisq ;
run;

data h.matching_high5;
set h.matching_high4;
if trtm=1 then ps1=ps;
if trtm=0 then ps0=ps;
run;

data h.matching_high6;
set h.matching_high5;
if trtm=1 then ps1=1;
if trtm=0 then ps0=0;
run;
data h.matching_high7;
set h.matching_high6;
if trtm=1 then ps1= logit;
if trtm= 0 then ps0= logit;
run;

proc sgplot data=h.matching_high7;
density ps0/legendlabel="Control";
density ps1/legendlabel=" case ";
title 'distribution of PS in matched sample';
label PS0='propensity score' PS1='propensity score'; keylegend; run;
```

Dataset	
h.Cohort_high, h.matching_high	As example, dataset for high risk group has been applied. Dataset for each risk group should be applied separately.
Categorical variables	
Tepisode	Treatment episode identifier. Treatment episode that occurred 180 days after the end of the previous episode was considered as a new episode
sex	Male=0, Female=1
age_g	65–74 years = 0, 75–84 year = 1, ≥85 years = 2
History of GI events	PUD_svr (history of GI bleeding or perforation), PUD_nonsvr (history of GI events other than bleeding or perforation)
Comorbidities	Cirrhosis, DM (Diabetes mellitus), CKD (Chronic kidney disease), HemorrS (Hemorrhagic stroke), cancer, alcoholism, GERD (Gastroesophageal reflux disease), IBD (Inflammatory bowel disease), tpenia (Thrombocytopenia), HF (Heart failure), HTN (Hypertension)
Co-medications	ACO[a] (Anticoagulant), APLT[b] (Antiplatelet), Steroid, SSRI (Selective serotonin reuptake inhibitors), LASA (Low dose aspirin), PYI (P2Y12 inhibitor)
RFcount	Number of risk factors
CCI_g	CCI score 0–1 = 0, CCI score 2–4 = 1, CCI score ≥5 = 2

Notes: [a]Anticoagulants include vitamin K antagonists, direct oral anticoagulants, unfractionated heparin, and low molecular weight heparin. [b]Antiplatelets include low dose acetylsalicylic acid, P2Y12 inhibitors, and phosphodiesterase inhibitors.

Abbreviations: ACO, anticoagulant; APLT, antiplatelet; CCI, Charlson comorbidity index; CKD, chronic kidney disease; DM, diabetes mellitus; GERD, gastroesophageal reflux disease; GI, gastrointestinal; GPA, gastroprotective agent; HF, Heart failure; HTN, Hypertension; IBD, inflammatory bowel disease; LASA, Low dose aspirin; PUD, peptic ulcer disease; PYI, P2Y12 inhibitor; SSRI, selective serotonin reuptake inhibitors.

Supplementary Table S2 ICD-10 code used for outcome and comorbidities

Disease	ICD-10 code
Outcome	
Serious GI complication (bleeding, perforation, obstruction)	K25.0, K25.1, K25.2, K25.4, K25.5, K25.6, K26.0, K26.1, K26.2, K26.4, K26.5, K26.6, K27.0, K27.1, K27.2, K27.4, K27.5, K27.6, K28.0, K28.1, K28.2, K28.4, K28.5, K28.6, K31.5, K56.6, K63.1, K92.0, K92.1, K92.2
Comorbidities	
Non-serious peptic ulcer disease (other than bleeding or perforation)	K25, K25.3, K25.7, K25.9, K26, K26.3, K26.7, K26.9, K27, K27.3, K27.7, K27.9, K28, K28.3, K28.7, K28.9, K63.3
Cirrhosis	K70.3, K71.7, K74.3, K74.4, K74.5, K74.6, K76.1, K78.8, K86.8
Diabetes mellitus	E08, E09, E10, E11, E13
Chronic kidney disease	N18, E08.22, E09.22, E10.22, E11.22, E13.22, I12, I13, Z94.0
Inflammatory bowel disease	K50.0, K50.1, K50.8, K50.9, K51.0, K51.2, K51.3, K51.4, K51.5, K51.8, K51.9
Hemorrhagic stroke	I60, I61
Cancer	C00, C01, C02, C03, C04, C05, C06, C07, C08, C09, C10, C11, C12, C13, C14, C15, C16, C17, C18, C19, C20, C21, C22, C23, C24, C25, C26, C30, C31, C32, C33, C34, C37, C38, C39, C40, C41, C45, C46, C47, C48, C49, C50, C51, C52, C53, C54, C55, C56, C57, C58, C60, C61, C62, C63, C64, C65, C66, C67, C68, C69, C70, C71, C72, C73, C74, C75, C76, C77, C78, C79, C7A, C7B, C80, C81, C82, C83, C84, C85, C86, C88, C90, C91, C92, C93, C94, C95, C96
Alcoholism	E24.4, F10, G31.2, G62.1, G72.1, I42.6, K29.2, K70, K86.0, O35.4, P04.3, Q86.0, T51, Y90, Z71.4
Gastroesophageal reflux disease	K21, K22.1
Thrombocytopenia	D69.1, D69.3, D69.4, D69.41, D69.42, D69.49, D69.5, D69.51, D69.59, D69.6
Heart failure	I50
Hypertension	I10, I11, I12, I13, I14, I15, O10, O11, O13, O16, P29.2

Abbreviations: GI, gastrointestinal; ICD, International Classification of Diseases.

Supplementary Table S3 Multivariate analysis of NSAID-induced serious GI complications in high risk group

Characteristics	aHR	95% CI
Age, years		
65–74	reference	reference
75–84	1.28	(0.53, 3.07)
≥85	7.75	(3.17, 18.93)
Sex, male	1.22	(0.58, 2.57)
Co-administration of GPA		
Rebamipide vs PPI[a]	2.63	(1.24, 5.59)
Comorbidities		
History of GI bleeding or perforation	5.25	(0.61, 45.39)
History of GI events other than bleeding or perforation	0.93	(0.42, 2.07)
Cirrhosis	0.00	(0, -)
Diabetes mellitus	1.53	(0.69, 3.41)
Chronic kidney disease	3.19	(0.90, 11.28)
Hemorrhagic stroke	0.00	(0, -)
Cancer	0.46	(0.06, 3.72)
Alcoholism	0.00	(0, -)
Gastroesophageal reflux disease	0.95	(0.45, 2.00)
Inflammatory bowel disease	0.00	(0, -)
Thrombocytopenia	0.00	(0, -)
Heart failure	0.78	(0.26, 2.36)
Hypertension	0.47	(0.19, 1.16)
Co-medications		
Anticoagulant	5.92	(1.96, 17.86)
Antiplatelet	2.67	(1.10, 6.50)
Steroid	1.70	(0.57, 5.09)
SSRI	11.00	(1.12, 107.90)
Number of risk factors		
1	reference	reference
2	0.62	(0.11, 3.36)
3	0.51	(0.04, 6.23)
4	0.00	(0, -)
CCI score		
0–1	reference	reference
2–4	1.79	(0.48, 6.70)
≥5	1.42	(0.32, 7.27)

Notes: [a]PPI served as the reference.

Abbreviations: aHR, adjusted hazard ratio; CCI, Charlson comorbidity index; CI, confidence interval; GI, gastrointestinal; GPA, gastroprotective agent; NSAID, nonsteroidal anti-inflammatory drug; PPI, proton-pump inhibitor; SSRI, selective serotonin reuptake inhibitor.

Supplementary Table S4 Multivariate analysis of NSAID-induced serious GI complications in moderate risk group with 2 risk factors

Characteristics	aHR	95% CI
Age, years		
65–74	reference	reference
75–84	1.35	(0.67, 2.74)
≥85	3.98	(1.70, 9.34)
Sex, male	2.12	(1.13, 3.96)
Co-administration of GPA		
Rebamipide vs PPI[a]	2.42	(1.21, 4.83)
Comorbidities		
History of GI bleeding or perforation	N/A	N/A
History of GI events other than bleeding or perforation	0.55	(0.23, 1.32)
Cirrhosis	6.32	(0.83, 47.96)
Diabetes mellitus	0.54	(0.25, 1.18)
Chronic kidney disease	0.77	(0.10, 5.82)
Hemorrhagic stroke	0.00	(0, -)
Cancer	0.64	(0.14, 2.91)
Alcoholism	0.00	(0, -)
Gastroesophageal reflux disease	1.37	(0.70, 2.66)
Inflammatory bowel disease	0.00	(0, -)
Thrombocytopenia	0.00	(0, -)
Heart failure	1.32	(0.45, 3.88)
Hypertension	0.76	(0.35, 1.64)
Co-medications		
Anticoagulant	1.41	(0.37, 5.40)
Antiplatelet	0.82	(0.40, 1.69)
Steroid	1.37	(0.40, 4.63)
SSRI	28.33	(3.04, 263.78)
CCI score		
0–1	reference	reference
2–4	3.25	(1.22, 8.69)
≥5	4.04	(0.98, 16.69)

Notes: [a]PPI served as the reference.

Abbreviations: aHR, adjusted hazard ratio; CCI, Charlson comorbidity index; CI, confidence interval; GI, gastrointestinal; GPA, gastroprotective agent; N/A, not applicable; NSAID, nonsteroidal anti-inflammatory drug; PPI, proton-pump inhibitor; SSRI, selective serotonin reuptake inhibitor.

Supplementary Table S5 Multivariate analysis of NSAID-induced serious GI complications in elderly patients without risk factors

Characteristics	aHR	95% CI
Age, years		
65–74	reference	reference
75–84	3.07	(1.17, 8.06)
≥85	1.90	(0.38, 9.58)
Sex, male	1.11	(0.45, 2.77)
Co-administration of GPA		
Rebamipide vs PPI[a]	0.69	(0.27, 1.76)
Comorbidities		
History of GI bleeding or perforation	N/A	N/A
History of GI events other than bleeding or perforation	N/A	N/A
Cirrhosis	0.00	(0, -)
Diabetes mellitus	0.76	(0.28, 2.04)
Chronic kidney disease	1.24	(0.15, 9.93)
Hemorrhagic stroke	5.44	(0.70, 42.16)
Cancer	0.00	(0, -)
Alcoholism	4.13	(0.52, 32.75)
Gastroesophageal reflux disease	0.80	(0.31, 2.02)
Inflammatory bowel disease	0.00	(0, -)
Thrombocytopenia	0.00	(0, -)
Heart failure	1.53	(0.35, 6.68)
Hypertension	1.48	(0.45, 4.88)
Co-medications		
Anticoagulant	N/A	N/A
Antiplatelet	2.72	(1.07, 6.93)
Steroid	N/A	N/A
SSRI	0.00	(0, -)
CCI score		
0–1	reference	reference
2–4	2.08	(0.59, 7.34)
≥5	2.02	(0.31, 13.27)

Notes: [a]PPI served as the reference.

Abbreviations: aHR, adjusted hazard ratio; CCI, Charlson comorbidity index; CI, confidence interval; GI, gastrointestinal; GPA, gastroprotective agent; N/A, not applicable; NSAID, nonsteroidal anti-inflammatory drug; PPI, proton-pump inhibitor; SSRI, selective serotonin reuptake inhibitor.