A Macro System for Standardizing and Grading Laboratory Data

Bruce Coate and William Hoskins
Agouron Pharmaceutical, Inc., San Diego, CA

ABSTRACT

In nearly all clinical trials involving humans it is necessary to collect and evaluate lab data. Frequently this evaluation consists of assigning a grade to each result, and then summarizing changes over time for each test. The purpose of this paper is to present an overview of how lab data is standardized and graded using SAS macros. The procedure outlined here starts at the point where the lab data has been entered into a database (Oracle Clinical in this case) and validated, and ends with the creation of a standardized, graded SAS dataset.

METHODS

Because laboratory data as received from both local and central labs is often ungraded, it may be useful to have a system of macros set up which can assign grades to lab results and that can be used from study to study. Such an application is described in this paper. The application consists of three macros: %convunit, %ctctolab, and %makelab.

Before the lab macros can be called, it is first necessary to create a SAS dataset containing the toxicity criteria as specified in the study protocol. This dataset is used later on by the %ctctolab macro to assign the grade to a particular test result. There are essentially two ways to assign grades: 1) using an absolute range, for example the grades for hyponatremia might be assigned as follows: grade 1 = 146-150 meq/L, grade 2 = 151-157 meq/L, grade 3 = 158-165 meq/L, grade 4 = <165 meq/L; 2) using a range relative to normal limits; the nci variable defined in the toxicity dataset below is used in the %ctctolab macro as a factor that is multiplied by the normal low and high ranges. These two different scenarios are defined in the toxicity dataset with the following code.

```
data toxicity;
delta = 0.0000000001;
high = 99999999;
low = -high;
* test specific criteria follow *;
Toxicity = 'Hypernatremia';
testname='Sodium Hi';
ctctype='H';
test = 'SODIUM';
unitnci='mEq/L';
grade=.; limit='HIGH';nci= .; output;
grade=.; limit='LOW '; nci= .; output;
grade=0; limit='HIGH'; nci=146-delta; output;
grade=0; limit='LOW '; nci=135+delta; output;
grade=1; limit='LOW '; nci=146; output;
grade=1; limit='HIGH'; nci=150; output;
grade=2; limit='HIGH'; nci=157; output;
grade=2; limit='LOW '; nci=150+delta; output;
grade=3; limit='HIGH'; nci=165; output;
grade=3; limit='LOW '; nci=157+delta; output;
grade=4; limit='HIGH'; nci=high; output;
grade=4; limit='LOW ' ; nci=165+delta; output;
lparm = 'BILIRUBN'; unitnci='ULN';
testname='Total Bilirubin'; Toxicity = 'Hyperbilirubinemia';
ctctype=' '; grade=.; limit='HIGH';nci= .; output;
grade=.; limit='LOW '; nci= .; output;
grade=0; limit='HIGH'; nci=1.1-delta; output;
grade=0; limit='LOW '; nci=low; output;
grade=1; limit='HIGH'; nci=1.5; output;
grade=1; limit='LOW '; nci=1.1; output;
grade=2; limit='HIGH'; nci=2.5; output;
grade=2; limit='LOW '; nci=1.5+delta; output;
grade=3; limit='HIGH'; nci=5.0; output;
grade=3; limit='LOW ' ; nci=2.5+delta; output;
grade=4; limit='HIGH'; nci=high; output;
grade=4; limit='LOW ' ; nci=5+delta; output;
* more test criteria ... *;
```

It is important to note that if a lab test is not specified in the protocol then it will not be assigned a grade. Also, grading can only be assigned for tests which are graded...
according to numeric ranges. For example, a toxicity such as proteinuria is not graded using this method since the grading criterion includes values such as 1+, 2+, etc. For such cases the site personnel must review the lab results and include aberrant values on adverse event case report forms.

Once the toxicity dataset has been set up the main macros are ready to be called. The order of the macro calls are shown below:

```
%makelab
   %ctctolab
   %convunit
```

The purpose of the %convunit macro is to convert all results into the same set of units. Due to space constraints the entire code is not given; however, one can get an idea of how the macro works by looking at the temperature, weight and height conversions.

```
%macro convunit(lo=, hi=, oldunit=);
  data &dsn(drop=factor);
  length factor 8;
  set &dsn;
  if (&oldunit in ('C' 'Celsius' 'Centigrade')) then do;
    if (&newunit in ('F' 'Fahrenheit')) then do;
      &lo = 1.8*&lo+32;
%if (&hi ^=) %then &hi = 1.8*&hi+32;;
      &oldunit=&newunit;
    end;
  end;
  else if (&oldunit in ('F' 'Fahrenheit')) then do;
    if (&newunit in ('C' 'Celsius' 'Centigrade')) then do;
      &lo = (&lo-32)/1.8;
%if (&hi ^=) %then &hi = (&hi-32)/1.8;;
      &oldunit=&newunit;
    end;
  end;
  else if (&oldunit in ('fL' 'um^3') and &newunit in ('fL' 'um^3')) then do;
    factor = 1.0;
  end;
  else if (&oldunit in ('Kilograms' 'KG') and &newunit in ('Pounds' 'LBS' 'LB')) then do;
    factor = 2.205;
  end;
  else if (&oldunit in ('Pounds' 'LBS' 'LB') and &newunit in ('Kilograms' 'KG')) then do;
    factor = 0.454;
  end;
  else if (&oldunit in ('Centimeters' 'CM') and &newunit in ('Inch' 'Inches' 'IN')) then do;
    factor = 0.394;
  end;
  else if (&oldunit in ('Inch' 'Inches' 'IN') and &newunit in ('Centimeters' 'CM')) then do;
    factor = 2.54;
  end;
  if (factor ^= .) then do;
    &lo=&lo*factor;
%if (&hi ^=) %then &hi=&hi*factor;;
    &oldunit = &newunit;
  end;
run;
```

*additional conversions would go here*;

```
proc print;
  where (&lo ^= . and (compress(&oldunit) ^= compress(&newunit)) and (&oldunit ^= in ('LLN' 'ULN')));
proc append base=master force
data=&dsn(keep=&lo &oldunit &newunit &testname
  where=(&lo ^= . and (compress(oldunit) ^= compress(newunit))
and (&oldunit ^= in ('LLN' 'ULN')))
rename=(&oldunit=oldunit
&newunit=newunit &lo=lo));
run;
```

```
proc summary data=master nway;
  class oldunit newunit &testname;
  output out=master(keep=oldunit
newunit &testname);
run;
```

*keep variable name of old unit;* data master; set master; voldunit="&oldunit"; run; %mend convunit;

So to convert the result units in the toxicity dataset to the standard units the macro call would look like %convunit(lo=nci,
oldunit=unitnci). The %convunit macro is also used to convert results in the lab dataset to the standard units.

The final macro that is needed by %makelab is %ctctolab. The %ctctolab macro is the heart of the system since it is here that the lab dataset and the toxicity dataset are combined, and the resulting grades are assigned.

%macro ctctolab
dsn=NORMLAB2,test=lparm,result=lvalstd,
rawreslt=lvalue,normlo=stdlorng,normhi=stdhiring,normunit=stunit);
* standardize units and testnames;
*get valid result unit for each test;
proc sort data = &dsn(keep=&test &normunit) nodupkey out=units;
where (&normunit is not missing);
by &test &normunit;
run;

*check and report if more than 1 unit for each test;
data units;
set units end=eof;
by &test &normunit;
if ^=first.&test and last.&test then do;
file print;
  put 'Multiple units for ' &test ' : ' &normunit;
end;
if eof then
  file log;
run;

*determine which version of the toxicity criteria to use based on study;
data _null_: set &dsn(keep=study);
if study in ('006', '009', '014') then do;
call symput('version', 'ctc_v20');
stop; end;
else if study in ('534', '502', '021') then do;
call symput('version','actgtc');
stop; end;
run;

proc sort data =
library.&version(rename=(lparm=&test))
out=ncitoxic;
by &test;
run;

* add result units to toxicity data;
data ncitoxic;
merge ncitoxic(in=in1) units(in=in2);
by &test;
if in1 and in2;
run;

*convert NCI criteria to result units;
%unitconv(dsn=ncitoxic,testname=&test,
normunit=&normunit);

*combine result and NCI Toxicity Criteria matching by test;
in order to conserve work space requirements break up hematology/chem joining;
* Note: &_patkey_ is a macro variable containing the patient identifier values *
proc sql;
create table _chem_ as select * from &dsn(where=(labcateg='C')) left join ncitoxic on &dsn..&test = ncitoxic.&test order by %do i = 1 %to %nwords(&patkey); %scan(&patkey_.&i), %end; colldt, colltm, &test, grdtype, grade, &result;
create table &dsn as select * from &dsn(where=(labcateg^='C')) left join ncitoxic on &dsn..&test = ncitoxic.&test order by %do i = 1 %to %nwords(&patkey); * %nwords will return the number of words in the string &patkey_ * %scan(&patkey_.&i), %end; colldt, colltm, &test, grdtype, grade, &result;
quit;

data &dsn;
set &dsn _chem_;
by &patkey_ colldt colltm &test grdtype grade &result;
run;

data &dsn(drop=unitnci);
set &dsn;
by &patkey_ colldt colltm &test grdtype grade &result;
*convert relative NCI criteria to absolute;
if (unitnci = 'LLN') then nci=nci*&normlo;
else if (unitnci = 'ULN') then nci=nci*&normhi;
*merge (transpose) lower and upper NCI limits;
data limitdsn (drop=limit);
  merge &dsn (rename = (nci=lownci) where
  = (limit='LOW')) &dsn (rename = (nci = hinci)
  where = (limit='HIGH'));
  by &_patkey_ colldt colltm &test grdtype
  grade &result;
run;

*identify assessments where both corrected and uncorrected Calcium exist and ungrade
the uncorrected Calciums;
data _corcal_
  set &dsn (keep = &_patkey_ colldt colltm
  &test where = (test = ' CORRCALC'));
  by &_patkey_ colldt colltm &test;
  lparm = 'CALCIUM';
run;

*combine data with and without nci limits;
data &dsn(drop=nci where=(limit is missing));
  merge limitdsn _corcal_(in=incorr)
  &dsn(drop=limit);
  by &_patkey_ colldt colltm &test;
  "if both corrected and uncorrected Calcium exist and ungrade the uncorrected
Calciums;
  if (incorr) then do;
    grade=.; testname=''; toxicity=''; lownci=.;
    hinci=.;
  end;
  " keep only if test has no NCI criteria or
  when normal ranges are provided and
  results fall within toxicity limits or
  normal ranges are
  not provided (keep only the observation with
  missing grade);
  if (testname = ' ' or (&normlo ^= . and
  lownci le &result le hinci) or (&normlo=. and
  grade =."lt 0"));

  "create not evaluable grade of -1 for tests
  having toxicity criteria and missing grade
  that have raw results;
  if (grade=. and testname ^= ' ' and
  &rawreslt ^= in ("',';ND';N.D.';--';--) ) then
    grade=1;
  format grade grade.;
run;

"if there are multiple grades for a result then
choose grade 0 over all others;

*this expected to occur for Lymphocytes
where grades 1 & 2 overlap with the normal limits;
proc sort data =&dsn;
  by &_patkey_ actevent colldt colltm &test
  &rawreslt grade grdtype;
run;

proc sort data =&dsn nodupkey;
  by &_patkey_ actevent colldt colltm &test
  &rawreslt;
run;

%mend cctctolab;

Now that all of the pieces are in place you
can call the main macro, %makelab. The
primary purpose of the %makelab macro is
to act as a driver for the other code
segments. In addition, it also includes code
to compute neutrophil counts if the bands
and segments are available.

%macro
makelab(dsnin=extract.normlab2,dsnout=no
rmlab2);
  libname libunit
    "&CLINPATH/software/sascats/preferred_un
its";
  proc sort data = &dsnin out=normlab2;
    by &_patkey_ actevent dcmname;
  run;
  *convert normal ranges (in labunit) to
  preferred lab units(stdunit);
  %unitconv(dsn=normlab2,testname=lparm,n
  ewunit=stdunit);
  data bandseg(drop=dummy dummy2
  lvalband lvalsegs stdlband stdhband
  stdlsags stdhsags stduband stdusegs);
    merge normlab2(where=(dummy='BANDS'
        and dummy2 ^= in ("',';ND';N.D.';--';--")
        rename=(lparm=dummy
          lvalstd=lvalband lvalue=dummy2
          stdlorng=stdlband stdhirng=stdhband
          stdunit=stduband) in=inband)
    normlab2(where=(dummy='SEGS' and
data bandseg(drop=dummy dummy2
  lvalband lvalsegs stdlband stdhband
  stdlsags stdhsags stduband stdusegs);
    merge normlab2(where=(dummy='BANDS'
        and dummy2 ^= in ("',';ND';N.D.';--';--")
        rename=(lparm=dummy
          lvalstd=lvalband lvalue=dummy2
          stdlorng=stdlband stdhirng=stdhband
          stdunit=stduband) in=insegs)
CONCLUSION

The SAS macros presented in this paper give an example of how it is possible to use macros to standardize and grade laboratory results. The same macros can be used from study to study as long as the appropriate toxicity dataset is first created by the user.