PharmaSUG 2016 - Paper QT13

Lab CTCAE – the Perl Way

Marina Dolgin, Teva Pharmaceutical Industries Ltd., Netanya, Israel

ABSTRACT

The NCI Common Terminology Criteria for Adverse Events (CTCAE) is a descriptive terminology which is utilized for Adverse Event (AE) reporting. A toxicity grading scale is provided for each AE term, it varies from 1 (mild) to 5 (death).

Typically, CTCAE grading is directly collected from the site on the adverse experience case report form. However, this may not be the case for laboratory results. Usually, the calculation of CTCAE grading for laboratory results would require an explicit coding of each criteria, per each lab test, by means of "If" and "Else" statements. This paper demonstrates a method (macro) for calculation of toxicity grade based on laboratory value and original NCI CTCAE grading file. This macro is using Perl Regular Expressions patterns and functions, which is much more compact solution to a complicated hard coding task and also eliminates a need for per test programming of the toxicity grades. Lastly, macro may be used with both ADaM and SDTM structures.

INTRODUCTION

The Common Terminology Criteria for Adverse Events (CTCAE) are a set of criteria for recognition and grading toxicity of adverse events (AE). The CTCAE system is a product of the US National Cancer Institute (NCI). It uses a range of grades from 1(Mild Adverse Event) to 5 (Death due to Adverse Event), when grade 0 considered as normal condition.

For adverse events CTCAE toxicity grading is directly collected from the site on the case report form (CRF). However, for laboratory values toxicity grading is assigned in later step by a clinical programmer.

CTCAE v4.0 Term	Grade 1	Grade 2	Grade 3	Grade 4

Anemia	Hemoglobin (Hgb)
	<LLN - 10.0 g/dL;
	<LLN - 6.2 mmol/L;
	<LLN - 100 g/L
		Hgb <10.0 - 8.0 g/dL;
		<6.2 - 4.9 mmol/L;
		<100 - 80g/L
			Hgb <8.0 g/dL;
			<4.9 mmol/L; <80 g/L;
			transfusion indicated
				Life-threatening consequences; urgent intervention indicated

Blood bilirubin increased	>ULN - 1.5 x ULN	>1.5 - 3.0 x ULN	>3.0 - 10.0 x ULN	>10.0 x ULN
Lymphocyte count decreased	<LLN - 800/mm3; <LLN - 0.8 x 10e9/L	<800 - 500/mm3; <0.8 - 0.5 x 10e9 /L	<500 - 200/mm3; <0.5 - 0.2 x 10e9 /L	<200/mm3; <0.2 x 10e9 /L

Table 1. Example of CTCAE v4.03.

Usually, the calculation of CTCAE grading for laboratory results would require an explicit coding of each criteria, per each lab test, as shown below for "Lymphocyte" test:

```
if lbtestcd ='LYMPH' and upcase(lbstresu)='10E9/L' then do;
   if 0.8<=lbstresn<lbstnrlo    then lbtoxgr='1';
     else if 0.5<=lbstresn<0.8  then lbtoxgr ='2';
       else if 0.2<=lbstresn<0.5 then lbtoxgr ='3';
         else if .z<lbstresn<0.2   then lbtoxgr ='4';
           else if lbstresn>.z then lbtoxgr ='0';
end;
```

Some limitations of explicit coding are:

1. Manual coding of all grades criteria found in CTCAE file, per test, per different units.
2. Increased chances for typo errors during hard coding.
3. The code should be revised in order to match different CTCAE versions.

The %lb_CTCAE macro is using Perl Regular Expressions patterns and functions that allow to locate patterns in text strings i.e. the toxicity criteria and to compare each grade criteria with laboratory value in the data.

Advantages of %lb_CTCAE macro:

1. No hard coding of grades per each test and units.
2. Values are extracted directly from original CTCAE file.
3. No modification is needed for different CTCAE MedDRA version (4.x & 3.x).
4. Might be used with both ADaM and SDTM structures.

CTCAE_v3.xls and CTCAE_v4.xls files are available on NCI website. As CTCAE terms do not always directly indicate the laboratory test to which the grades apply there is a need to assign a testcode to each criteria before working with that file. In addition transposition of the file by testcode and grade is performed. These operations are done only once when a new CTCAE MedDRA version is released.

MACRO DESIGN

```
%lb_CTCAE(ctc=, datain=, dataout=, tests=, isAdam=)

&ctc     -CTCAE.xls file location
&datain  -input LAB dataset
&dataout -output dataset
&tests   -tests of interest for which grade is calculated
&isAdam  -'Y' for ADaM structure; else SDTM.
```

PREPROCESSING STEP

As macro works on both ADaM and SDTM structure, the correct variables from data should be used.

```
%if &isAdaM="Y" %then %do;
       %let lab_value =aval;
       %let high_lim =a1hi;
       %let low_lim =a1lo;
%end;
%else %do;
      %let lab_value =lbstresn;
       %let high_lim =lbstnrhi;
       %let low_lim =lbstnrlo;
%end;
```

In first step only tests of interest are taken from laboratory data. In addition, the sequence variable SEQ is created.

```
data lb_test;
set &datain.;
if testcode in (&tests.);
seq =_n_;
run;
```

Subsequent merging of metadata file CTCAE.xls with laboratory data is performed. The merging is implemented by testcodes -found in both metadata and lab data.

EXTRACTION OF THE RANGES FROM CTCAE FILE

The regular expression is a sequence of characters that define a search pattern. Once a pattern is found, it's possible to obtain the position of the pattern, extract a substring, or substitute a string. There are two main patterns of toxicity ranges in CTCAE guide. The first pattern contains units (e.g. "Anemia"); the second pattern is a product of the numerical coefficient and upper (ULN) or lower (LLN) limit of the normal range, without units (e.g. "Blood bilirubin increased ", Table 1).

RANGES WITH UNITS

An extraction of the ranges for laboratory tests based on standard units found in data (e.g. Hemoglobin in g/L).

```
data CTCAE_lab;
set CTCAE_lab;
if lbstresu ^= '' then do;
     pattern = CAT("#(<|>)[a-z0-9-.,]+",lbstresu,"#I"); /*Note 1*/
     ptrn_ = PRXPARSE(pattern);
     CALL PRXSUBSTR(ptrn_,descrp,start,length);
     range_ = SUBSTRN(descrp,start,length);
end;
```

Note 1. Short explanation of regular expression pattern:

- Hash sign "#" is the delimiter that's specifying a pattern to search.
- The pattern starts with either *less* or *greater* symbol, followed by "[a-z0-9-.,]" - any letter or number, might contain a point or a comma, this part can be repeated one or more time - "+".
- The CAT function adds standard units from the data to the pattern.
- The "I" after the final delimiter is used in order to perform case-insensitive search.

The PRXPARSE function is used to create a regular expression.
The PRXSUBSTR call finds the correct range, the length of it and starting position; subsequent SUBSTRN function extracts the actual range from the text string.

At this point RANGE_ variable contains correct toxicity grade limits based on units that are found in data. Next step would be removal of units from the RANGE_ variable.

```
pattern2= CAT("s#(x?", lbstresu,")# #I"); /*Note 2*/
ptrn = PRXPARSE(pattern2);
range = PRXCHANGE(ptrn,-1,range_);
```

Note 2. Short explanation of regular expression pattern2:

- This pattern is used with PRXCHANGE function in order to remove units from range.
- The "s" in the beginning of the pattern specifies the substitution operator.
- The pattern itself might "?" contain the multiply symbol "x" and is concatenated with standard units.
- "# #" Is the substitute portion of the regular expression, it's blank and therefore PRXCHANGE function will remove the identified pattern (standard units) from the string.
- PRXCHANGE function substitutes units in RANGE_ with blank.

The last step could be possibly be done by usage of e.g. TRANWRD function; however PRXCHANGE together with regular expression allow to cover cases where "x" sign is found in the grade description before units like in "Lymphocyte count decreased" (Table 2). At the end of this step, the RANGE variable contains grade toxicity limits without units.

Testcode	ADVERSE_EVENT	GRADE	GRADE_DESCRIPTION	Standard Units	RANGE_	RANGE

T046	Anemia	1	Hemoglobin(Hgb)
			<LLN-10.0g/dL;
			<LLN-6.2mmol/L;<LLN-100g/L
				g/L	<LLN-100g/L	<LLN-100

T046	Anemia	2	Hgb<10.0-8.0g/dL; <6.2-4.9mmol/L;<100-80g/L	g/L	<100-80g/L	<100-80

T046	Anemia	3	Hgb<8.0-6.5g/dL;
			<4.9-4.0mmol/L;
			<80-65g/L;transfusion indicated
				g/L	<80-65g/L	<80-65

T046	Anemia	4	Hgb<6.5g/dL;<4.0mmol/L;<65g/L;	g/L	<65g/L	<65
T402	Lymphocyte count decreased	1	<LLN - 800/mm3; <LLN - 0.8 x 10e9/L	10E9/L	<LLN-0.8x10e9/L	<LLN-0.8
T402	Lymphocyte count decreased	2	<800 - 500/mm3; <0.8 - 0.5 x 10e9 /L	10E9/L	<0.8-0.5x10e9/L	<0.8-0.5
T402	Lymphocyte count decreased	3	<500 - 200/mm3; <0.5 - 0.2 x 10e9 /L	10E9/L	<0.5-0.2x10e9/L	<0.5-0.2
T402	Lymphocyte count decreased	4	<200/mm3; <0.2 x 10e9 /L	10E9/L	<0.2x10e9/L	<0.2

Table 2. Extraction of the ranges based on standard units.

RANGES WITHOUT UNITS

For criteria that are based on the products of upper or lower bound of the normal range next step is being applied in order to calculate toxicity ranges (see Table 3). Here the multipliers are of interest, therefore first step will be detection of multiplier by regular expression pattern, and then multiplication of ULN/LLN by detected value.

```
%let digit = \d+\.?\d*;    /* multiplier digit */
data CTCAE_lab2;
set CTCAE_lab;
   retain pattern3 pattern4 pattern5;
   if _N_ = 1  then do;
           pattern3 = PRXPARSE("#>(ULN)-(&digit.)x(ULN)#"); /*Note 3*/
           pattern4 = PRXPARSE("#>(&digit.)-(&digit.)x(ULN)#");
           pattern5 = PRXPARSE("#>(&digit.)x(ULN)#");
       end;
       if find (descrp, 'x') > 0 then do;
             if PRXMATCH(pattern3,descrp) >0 then do;
                 CALL PRXPOSN(pattern3,2,start,length);
                 range = CATS(">",&high_lim., "-", SUBSTR(descrp,start,length)*
   &high_lim.) ;
             end;
             else if PRXMATCH(pattern4,descrp) >0 then do;
                 CALL PRXPOSN(pattern4,1,start,length);
                 CALL PRXPOSN(pattern4,2,start2,length2);
                 range = CATS (">",SUBSTR(descrp,start,length)*&high_lim., "-",
   SUBSTR(descrp,start2,length2)*&high_lim.);
             end;
             else if PRXMATCH(pattern5,descrp) >0 then do;
                    CALL PRXPOSN(pattern5,1,start,length);
                    range = CATS(">",SUBSTR(descrp,start,length)*&high_lim.);
             end;
       end;
   run;
```

The same approach applies also to LLN values.

Note 3. Short explanation of regular expression pattern3:

- &digit -is the multiplying value.
- The pattern starts with *greater* symbol, followed by `"(ULN)-(&digit.)x(ULN)"`.
- Each set of parentheses inside the regular expression represents a subexpression. The PRXPOSN function identifies position and length of a subexpression defined in the regular expression. As the purpose of this step is to identify multiplier value, the number of subexpression to be evaluated by PRXPOSN is 2.
- For pattern4 PRXPOSN uses 1 & 2 subexpressions and for pattern5 -1 subexpression.

At the end of this step, the RANGE variable contains grade toxicity limits after multiplication of ULN from laboratory data by multiplier from GRADE_DESCRIPTION variable.

Lab. Code	**ADVERSE_EVENT**	**GRADE**	**GRADE_DESCRIPTION**	**Reference Range Upper Limit-Std Units**	**RANGE**
T506	Blood bilirubin increased	1	>ULN-1.5xULN	17.1	>17.1-25.65
T506	Blood bilirubin increased	2	>1.5-3.0xULN	17.1	>25.65-51.3
T506	Blood bilirubin increased	3	>3.0-10.0xULN	17.1	>51.3-171
T506	Blood bilirubin increased	4	>10.0xULN	17.1	>171

Table 3. Calculation of the ranges without standard units.

TOXICITY GRADE CALCULATION

Finally, identification of toxicity grade by comparison of laboratory result value with each grade limits is performed. In general, each RANGE might be presented by one of the four possible patterns, PATTERN3-PATTERN6 below. After detection of the pattern, lower and upper bound values are being extracted from RANGE by PRXPOSN call, comparison of laboratory value (AVAL or LBSTRESN) to the limits is performed.

```
   data CTCAE_lab3;
   set   CTCAE_lab2;
   if _N_ = 1  THEN do;
      PATTERN3 = PRXPARSE("#>(&digit.)(-)(&digit.)#"); /* >x - y */
      PATTERN4 = PRXPARSE("#<(&digit.)(-)(&digit.)#"); /* <x - y */
      PATTERN5 = PRXPARSE("#<(&digit.)#"); /* <x */
      PATTERN6 = PRXPARSE("#>(&digit.)#"); /* >x */
   end;
   retain PATTERN:;

   if PRXMATCH(PATTERN3,range) >0 then do;
      CALL PRXPOSN(PATTERN3,3,start3,length3);
      upper_level = SUBSTR(range,start3,length3);
      upper_level2 = input (upper_level,best.);
      CALL PRXPOSN(PATTERN3,1,start3,length3);
      low_level = SUBSTR(range,start3,length3);
      low_level2 = input (low_level,best.);
       if low_level2 <&lab_value.<= upper_level2  then output; /* >x - y */
   end;

   else if PRXMATCH(PATTERN4,range) >0 then do;
      CALL PRXPOSN(PATTERN4,1,start3,length3);
      upper_level = SUBSTR(range,start3,length3);
      upper_level2 = input (upper_level,best.);
      CALL PRXPOSN(PATTERN4,3,start3,length3);
      low_level = SUBSTR(range,start3,length3);
      low_level2 = input (low_level,best.);
       if low_level2 <=&lab_value.< upper_level2  then output; /* <x - y */ end;

   else if PRXMATCH(PATTERN5,range) >0 then do;
      CALL PRXPOSN(PATTERN5,1,start3,length3);
      low_level = SUBSTR(range,start3,length3);
      low_level2 = input (low_level,best.);
       if low_level2 > &lab_value. then output;        /* <x */
   end;

   else if PRXMATCH(PATTERN6,range) >0 then do;
      CALL PRXPOSN(PATTERN6,1,start3,length3);
      low_level = SUBSTR(range,start3,length3);
      low_level2 = input (low_level,best.);
       if low_level2 < &lab_value. then output;      /* >x */
   end;
run;
```

MACRO OUTPUT

The last dataset contains only records that matched the PATTERN3- PATTERN6 i.e. have any toxicity grade. Those records are merged back to the original laboratory dataset by sequence variable, created in data preprocessing step. In addition, tests of interest, that don't have toxicity value, are getting grade "0". The final dataset – `&dataout` has two new variables: LBTOXGR (toxicity grade, e.g. "2") and LBTOX (AE description, e.g. "Anemia").

CONCLUSION

The %lb_CTCAE is using Perl Regular Expressions patterns and functions, and provide compact solution to a complicated hard coding task and eliminates a need for per test programming of the toxicity grades. In contrast to hard coding of toxicity grading, which would require a separate coding for each CTCAE version, the macro utilizes original CTCAE MedDRA file and therefore may be applied to different CTCAE versions without any code modifications. In addition, it designed to be used with both ADaM and SDTM structures.

REFERENCES

- National Cancer Institute, Common Terminology Criteria for Adverse Events (CTCAE) v4.03. evs.nci.nih.gov/ftp1/CTCAE/CTCAE_4.03_2010-06-14.xls
- Ron Cody, "An Introduction to Perl Regular Expressions in SAS 9", Robert Wood Johnson Medical School, Piscataway, NJ, SUGI29, Available at http://www2.sas.com/proceedings/sugi29/265-29.pdf

ACKNOWLEDGMENTS

I thank Svetlana Rubinchick for being my reviewer and offering great feedback.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name:	Marina Dolgin
Enterprise:	Teva Pharmaceutical Industries Ltd.
Address:	Hatrufa Street 12
City, State ZIP:	Netanya 42504, Israel
E-mail:	marina.dolgin@teva.co.il
Web:	www.tevapharm.com