SAS® IML Code To Calculate an Upper Confidence Limit for Multivariate Statistical Distance

Margaret Connolly, Wyeth Lederle Vaccines, Pearl River, NY

ABSTRACT

Pharmaceutical industry manufacturing process changes require documentation of similarity between products of pre-change and post-change processes. One test is tablet rate of dissolution: twelve tablets from each batch are dissolved under controlled conditions. The percent of tablet strength in solution from each tablet is measured at several time-points (5, 10, 15, 20 minutes). For highly variable products, similarity between profiles may be demonstrated by showing that the bounds on a confidence interval about the difference between lots are within the typical coefficient of variation for the original lot (e.g. within +/- 20%). The multivariate confidence interval for a profile with four time points is an 'ellipse' in four dimensions. To find the 'upper bound' in multivariate space is accomplished by Newton-Raphson iterations on a system of equations to define the confidence region and optimize distance. The accompanying code highlights the power and dimension independence of IML programming.

INTRODUCTION

A strategy to 'prove' equivalence between products is to estimate a 90% confidence interval about the mean difference for some key measurement and then compare the upper and lower bounds of the confidence interval to that true difference which is known, *a priori*, to have little practical effect. In particular, clinical trials conducted prior to approval of the pharmaceutical product may be used as evidence that typical batch to batch differences in the rate of dissolution and other potency attributes are consistent with equivalent clinical outcomes. Many immediate release products have only one or two dissolution time points measured below 85% and batch to batch differences are typically within 10%. Documentation for a post-approval change in the manufacturing process or site for such products must include evidence that the dissolution profile of material manufactured after the process change is not more than 10% different from that of a recent lot manufactured prior to the change. A recent publication (1) recommends a method for the less usual case to evaluate dissolution profiles with several time points and with moderately high within batch and batch to batch variability. This paper presents the SAS® IML code to implement this method.

SAMPLE DATA

Sample data from the reference (1) by Tsong et. al. are presented in Appendix A. The testing is non-destructive and Tablet 1 of the 5-minute time point is the same as tablet 1 of subsequent intervals. This design allows estimation of correlation among dissolution time points and is critical for the multivariate data analysis. The sample data include 6 tablets from each batch, although the standard protocol tests 12/batch.

As represented in Figure 1, dissolution profiles of the test and reference batches appear to be qualitatively different. The rate of dissolution is greater for tablets of the reference batch compared to the test batch before the 40-minute time point. After 40 minutes, the rate of dissolution appears to be greater for the test batch than for the reference batch.

The mean differences in percent of total tablet weight dissolved listed in Appendix A are generally large relative to estimates of the within-batch variation at each time point. Although the mean difference at each dissolution time point is statistically significant relative to within-batch variation, this has no relevance to the analysis: Similarity between the test and reference batches is evaluated relative to a critical limit for similarity, determined from an independent set of reference batches. In this example, the stated acceptance limit is 15%.

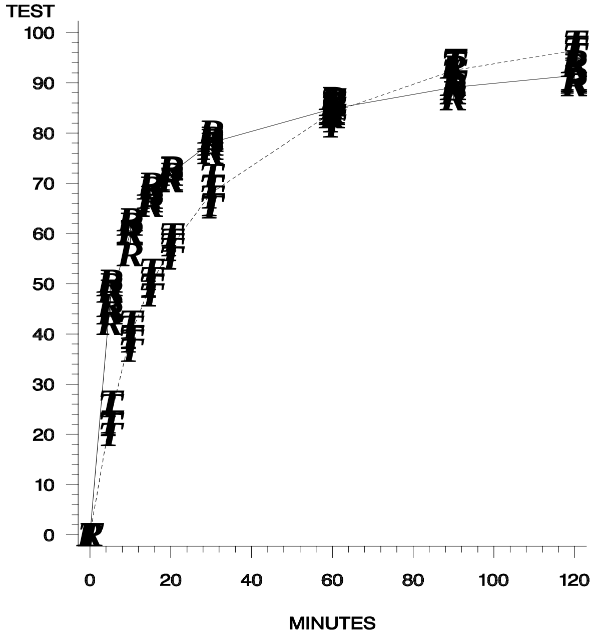

Figure 1. Dissolution Profile of Sample Data

UNIFORM LOCAL SIMILARITY OR GLOBAL SIMILARITY

Uniform local similarity would require that the batch difference of each evaluation interval is less than 15%. This condition is not met at the 5, 10, or 15 minute time points. The global similarity condition is met if two profiles at least 15% different at every time point are found significantly ($p<0.05$) more different than the sample batches. The test statistic for this comparison is the Mahalanobis distance (difference/'Standard.Error') of that point on the multivariate confidence region, which has maximum distance from the origin. If this is less than the corresponding distance for the vector which is uniformly 15% different at each time point then the hypothesis of dissimilar batches may be rejected.

THE MULTIVARIATE CONFIDENCE REGION

All points (y) within the multivariate confidence region satisfy the condition that $K(y' - d')V^{-1}(y-d) \le F_{P,\ 2P-P,90}$. For points on the boundary, this condition is an equality. The observed difference (d) is at the center of the confidence region. V is the pooled covariance matrix of the samples. For a single time point K=n/2: $V/K = (\text{standard error of mean difference})^2$. Generally $K= [(n^2)/(2n)]*(2n-P-1)/[(2n-2)P]$. The F distribution evaluated at the 90% level gives a 5% significance test of the one-sided hypothesis: The global difference is greater than or equal to 15%.

Figure 2 represents the multivariate confidence region projected on the plane defined by differences at 15 and 30 minutes when all other differences are fixed at the observed values. This region is inconsistent with a hypothesis that differences between the batches evaluated at the 15 and 30 minute intervals are bounded by 15%.

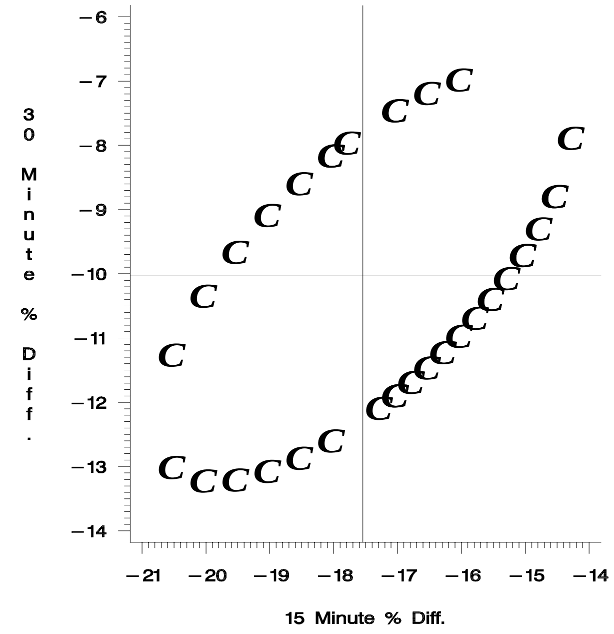

Figure 2. Confidence Region (90%) at (15,30) Minutes

Figure 3 represents the confidence region projected on the plane defined by differences at 60 and 90 minutes when all other differences are fixed at the observed values. This region is consistent with a hypothesis that differences between the batches evaluated at the 60 and 90 minute intervals are bounded by 15%.

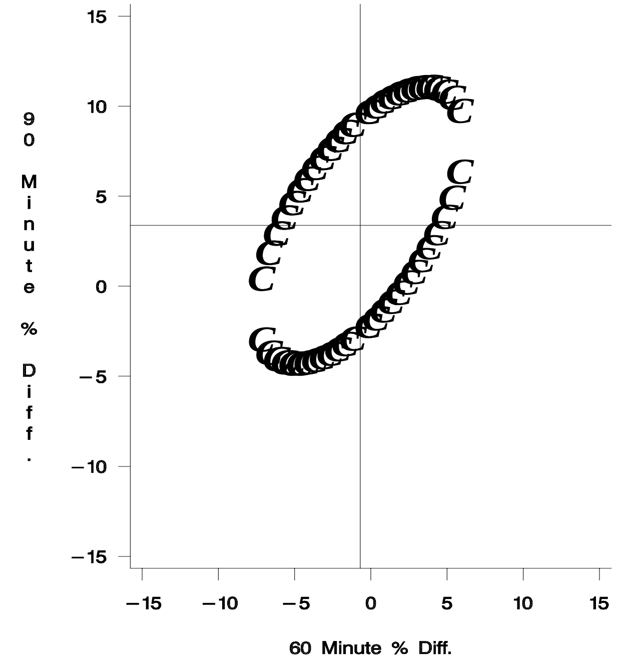

Figure 3. Confidence Region (90%) at (60,90) Minutes

CALCULATIONS

Comments within Appendix B include tips for applying this method. Program output, for comparison with worked examples in the reference (1) are:

RESULTS FOR TWO TIMEPOINTS

```
Mahalinobis Distance=10.4404 &  D^2= 109.00297
Degrees of Freedom =9     K factor= 1.35
F statistic=147.15401 Critical F(90%)= 3.006
Critical Limits
15%,   9.631
20%,  12.841
One Solution to Maximize Y`(V-Inverse)Y on CI
boundary
  -15.034
    2.903
On Confidence Bound kD'(V-1)D :   3.006
sqrt ( Y'(V-1)Y ) =   8.948
One Solution to Maximize Y`(V-Inverse)Y on CI
boundary
  -20.049
    3.871
On Confidence Bound kD'(V-1)D :   3.006
sqrt ( Y'(V-1)Y ) =  11.933
```

RESULTS FOR EIGHT TIMEPOINTS

```
Mahalinobis Distance=27.05430 & D^2=731.93498
Degrees of Freedom =3     K factor=   0.1125
F statistic=82.34268    Critical F(90%)=5.252
Critical Limits
15%,  29.561
20%,  39.415
One Solution to Maximize Y`(V-Inverse)Y on CI
boundary
          1. -17.708       5. -7.498
          2. -15.338       6. -0.521
          3. -13.112       7.  2.531
          4. -11.060       8.  3.736
On Confidence Bound kD'(V-1)D :   5.252
sqrt ( Y'(V-1)Y ) =  20.222
One Solution to Maximize Y`(V-Inverse)Y on CI
boundary
           1. -29.675    5. -12.565
           2. -25.702    6.  -0.873
           3. -21.972    7. 4.242
           4. -18.533    8. 6.261
On Confidence Bound kD'(V-1)D = 5.252
sqrt ( Y'(V-1)Y ) =  33.887
```

CONCLUSION

The method presented by Tsong et.al. has intuitive appeal for comparing profile plots. The method can be visualized by projecting onto the plane defined by differences at two time points. It appears that sample data was chosen to demonstrate that the method cannot get 'good' results from 'bad' data.

REFERENCE

1. Tsong Y, Hammerstrom T, Sathe P, Shah VP. Statistical Assessment of Mean Differences between Two Dissolution Data Sets, *Drug Info. J.* **30**:1105-1112 (1996).

ACKNOWLEDGMENTS

Code was written under contract to ProSoft, Inc. of Ambler, PA.

CONTACT INFORMATION

Margaret A. Connolly
Wyeth Lederle Vaccines
401 North Middletown Road, Building 140/421D
Pearl River, NY 10965
845 732-5859 (fax: 914 942 1166)
connolm3@war.wyeth.com

Appendix A. Dissolution Data of Test Batch Compared to Reference Batch

Batch	Tablet	5-min	10-min	15-min	20-min	30-min	60-min	90-min	120-min
Test	1	19.99	36.7	47.77	55.08	65.69	81.37	92.39	97.1
	2	22.08	39.29	49.46	56.79	67.22	82.42	89.93	95.62
	3	21.93	38.54	47.76	55.14	65.25	83.49	90.19	95.62
	4	22.44	39.46	49.72	58.67	69.21	84.93	94.12	95.51
	5	25.67	42.35	52.68	59.71	71.51	86.61	93.8	96.7
	6	26.37	41.34	51.01	57.75	69.44	85.9	94.45	98.07
	Mean	**23.08**	**39.61**	**49.73**	**57.19**	**68.05**	**84.12**	**92.48**	**96.44**
	StdDev	**2.44**	**2.01**	**1.90**	**1.88**	**2.42**	**2.04**	**2.00**	**1.04**
Reference	1	42.06	59.91	65.58	71.81	77.77	85.67	93.14	94.23
	2	44.16	60.18	67.17	70.82	76.11	83.27	88.01	89.59
	3	45.63	55.77	65.56	70.5	76.92	83.91	86.83	90.12
	4	48.52	60.39	66.51	73.06	78.54	84.99	88	93.43
	5	50.49	61.82	69.06	72.85	78.99	86.86	89.7	90.79
	6	49.77	62.73	69.77	72.88	80.18	84.2	88.88	90.47
	Mean	**46.77**	**60.13**	**67.28**	**71.99**	**78.09**	**84.82**	**89.09**	**91.44**
	StdDev	**3.35**	**2.40**	**1.78**	**1.12**	**1.47**	**1.31**	**2.20**	**1.91**
Test-Reference	Mean	-23.69	-20.52	-17.54	-14.80	-10.03	-0.70	3.39	5.00
Test-Reference	StdDev	4.15	3.13	2.61	2.19	2.83	2.42	2.98	2.17

Appendix B. SAS® Code to Evaluate Global Similarity

```
*** The file tsongdat has variables minutes, tablet, test, reference. Sorted by tablet;
proc transpose out=alltimes prefix=tabl data=tsongdat; by tablet;
   id minutes;
   var test referenc;
run;
data alltimes(drop=_name_); set alltimes;
    if _name_='TEST' then batch=1;
    else batch=2;
run;
proc sort data=alltimes; by batch; run;
************************************************************************;
***** Check on calculations                                        *****;
*****       The squared mahalinobis distance and F statistic       *****;
*****          are included in PROC DISCRIM output                 *****;
************************************************************************;
proc discrim data=alltimes method=normal
              pool=yes can distance manova noclassify;
  class batch;
  var tabl15 tabl90;
run;
************************************************************************;
***** The variable names represent time points                     *****;
*****       but calculations will use VN1, VN2, . . . VNP          *****;
************************************************************************;
proc corr cov noprint data=alltimes outp=pearson0; by batch;
  var TABL5 TABL10 TABL15 TABL20 TABL30 TABL60 TABL90 TABL120;
run;

************************************************************************;
***** A macro copied from "Writing Utility Macros"                 *****;
*****       chapter of V6 SASr Guide to Macro Processing           *****;
************************************************************************;
%macro strings(startat, upto, prefix);
```

```
   %local ii;
   %do ii = &startat %to &upto;
     &prefix&ii
   %end;
%mend strings;

%macro bound(number);

   proc iml;
%**********************************************************************;
%***** All data is read from pearson                             *****;
%***** All key output written to 'ofile'                         *****;
%***** Use RESET PRINT for debugging                             *****;
%**********************************************************************;
     use pearson;
     file ofile;
     reset noprint;
%**********************************************************************;
%***** The SIMULT module is called for each Newton-Raphson step   *****;
%*****    Recall that the parm+1 position represents 'lambda'     *****;
%*****    of lambda multiplier method to optimize under constraint ****;
%**********************************************************************;
     start simult;
       do ii = 1 to parms;
          vcol= vinverse[1:parms,ii];
          scorvec[ii]= 2*(1-k_factor*yvec[parms+1]) * yvec[1:parms]` * vcol
                     + 2 * k_factor*yvec[parms+1] * diffrenc` * vcol;
          do jj = 1 to ii;
             second[ii,jj] = 2*(1-k_factor*yvec[parms+1]) * vcol[jj];
          end;
          second[parms+1,ii]= -2*k_factor*yvec[1:parms]`*vcol+2*k_factor*diffrenc`*vcol;
       end;
       scorvec[parms+1] = -k_factor*(yvec[1:parms]-diffrenc)`*
                                       vinverse * (yvec[1:parms] - diffrenc) + fcrit;
       second[parms+1,parms+1]=0;
       do ii = 1 to parms;
         do jj= ii+1 to parms+1;
            second[ii,jj] = second[jj,ii];
         end;
       end;
     finish simult;
%**********************************************************************;
%***** Read in labels (minutes), var-cov matrices and means       *****;
%**********************************************************************;
    read all var { %strings(startat=1, upto=&number, prefix=const) } where(_type_='N' & batch=1);
      minutes = const1 %strings(startat=2, upto=&number, prefix=// const);
    read all var { %strings(startat=1, upto=&number, prefix=vn) } where(_type_='COV' & batch=1);
      s1= vn1 %strings(startat=2, upto=&number, prefix=|| vn);
    read all var { %strings(startat=1, upto=&number, prefix=vn) } where(_type_='COV' & batch=2);
       s2= vn1 %strings(startat=2, upto=&number, prefix=|| vn);
    read all var { %strings(startat=1, upto=&number, prefix=vn) } where(_type_='MEAN' & batch=1);
      m1= vn1 %strings(startat=2, upto=&number, prefix=// vn);
    read all var { %strings(startat=1, upto=&number, prefix=vn) } where(_type_='MEAN' & batch=2);
       m2= vn1 %strings(startat=2, upto=&number, prefix=// vn);
%**********************************************************************;
%***** All calculations based on observed mean difference and    ******;
%*****    pooled var-cov matrix (inverted)                       ******;
%**********************************************************************;
       diffrenc=m1-m2;
       spool = 0.5*(s1 + s2);
       vinverse=inv(spool);
%**********************************************************************;
%***** Print out some numbers to check vs. PROC DISCRIM output   ******;
%*****    and intermediate calculations given in reference       ******;
%**********************************************************************;
   put  "Observed Mean Difference ";
       do prti = 1 to nrow(minutes);
          put (minutes[prti]) 4.0 '  minutes, ' (diffrenc[prti]) 6.3;
       end;
       mahal = sqrt (diffrenc` * vinverse * diffrenc);
       d_square = mahal*mahal;
   put "Mahalinobis Distance =" (mahal) 12.5 "    and D^2=" (d_square) 12.5;
       read all var {vn1} where(_type_='N' & batch=1);
       ntablet = vn1;
       parms = round(trace( vinverse * spool ));
       df= ntablet + ntablet - parms - 1;
       k_factor =((ntablet*ntablet)/(2*ntablet))*
                        (2*ntablet - parms - 1)/((2*ntablet-2)*parms);
       ftest=k_factor*d_square;
   put "Degrees of Freedom =" (df) 3. "  K factor=" (k_factor) "   F statistic=" (ftest) 12.5;
       fcrit = finv(0.90, parms, df);
   put "Critical F(90%)=" (fcrit) 7.3;
%**********************************************************************;
%***** These are the critical upper limits at 10%, 15%, & 20%    ******;
%**********************************************************************;
   vec10 = j(parms,1)*10; lim10 = vec10` * vinverse * (vec10);  lim10 = sqrt(lim10);
   put "Critical Limits"; put "10%," (lim10) 8.3;

   vec15 = j(parms,1)*15; lim15 = vec15` * vinverse * (vec15);  lim15 = sqrt(lim15);
   put "Critical Limits"; put "15%," (lim15) 8.3;

   vec20 = j(parms,1)*20; lim20 = vec20` * vinverse * (vec20);  lim20 = sqrt(lim20);
   put "Critical Limits"; put "20%," (lim20) 8.3;

   vec25 = j(parms,1)*25; lim25 = vec25` * vinverse * (vec25);  lim25 = sqrt(lim25);
   put "Critical Limits"; put "25%," (lim25) 8.3;
%**********************************************************************;
%***** Initialize for Newton-Raphson Search                      ******;
%**********************************************************************;
       scorvec = j(parms+1,1);
       second = i(parms+1);
       yvec = j(parms+1,1);
       jvec =j(parms+1);
%**********************************************************************;
%***** Now try to find a point on the confidence boundary which  ******;
%*****   is at a Min or Max distance from (0,0,...0).            ******;
%*****   Recall from PROC NLIN documentation that N-R method is  ******;
%*****     easy to program but not robust.                       ******;
%**********************************************************************;
    do eval =1 to 50 while(abs(jvec` *scorvec) > 0.001) ;
       run simult;
       delta = solve(second,scorvec);
       yvec = yvec - delta;
    end;
    put "One Solution to Maximize Y`(V-Inverse)Y on CI boundary";
    do prti = 1 to nrow(minutes);
        put (yvec[prti]) 9.3 ;
       end;
%**********************************************************************;
%***** A check, if k*(D^2) is not equal to critical F then this  ******;
%*****   point is not on the confidence region boundary          ******;
%**********************************************************************;
       kdvd = k_factor * (yvec[1:parms] - diffrenc)` * vinverse * (yvec[1:parms] - diffrenc);
       put "On Confidence Bound kD'(V-1)D :" (kdvd) 8.3 "= Critical F:" (fcrit) 8.3;
%**********************************************************************;
%***** And if this is a solution, is the weighted distance the   ******;
%*****   Min or the Max, and, if Max                             ******;
%***** Is the weighted distance less than the critical limit?    ******;
%**********************************************************************;
       yvy = yvec[1:parms]` * vinverse * yvec[1:parms];
       sqryvy = sqrt(yvy);
       put "sqrt ( Y'(V-1)Y ) =" (sqryvy) 8.3;
%**********************************************************************;
%***** After finding Min or Max, try to fold across the observed ******;
%*****   difference and find the other limit                     ******;
%***** First re-set some values to indicate solution not found   ******;
%**********************************************************************;
```

```
       diffzero = diffrenc // 0;
       scorvec = j(parms+1,1);
       second = i(parms+1);
       jvec =j(parms+1);

%* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *;
%* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *;
%***** This ad hoc re-set works for these numbers.               *****;
%*****    the 'projection' would be yvec = 2*diffzero-yvec       *****;
%*****           i.e.  yvec = diffzero - (yvec-diffzero)         *****;
%* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *;
%* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *;
       yvec =   2* diffzero;
    do eval =1 to 50 while(abs(jvec` *scorvec) > 0.001) ;
       run simult;
       delta = solve(second,scorvec);
       yvec = yvec - delta;
    end;

    put "One Solution to Maximize Y`(V-Inverse)Y on CI boundary";
    do prti = 1 to nrow(minutes);
       put (yvec[prti]) 9.3 ;
    end;
%**********************************************************************;
%***** A check, if k*(D^2) is not equal to critical F then this  ******;
%*****   point is not on the confidence region boundary          ******;
%**********************************************************************;
       kdvd = k_factor * (yvec[1:parms] - diffrenc)` * vinverse * (yvec[1:parms] - diffrenc);
       put "On Confidence Bound kD'(V-1)D :" (kdvd) 8.3 "= Critical F:" (fcrit) 8.3;
%**********************************************************************;
%***** And if this is a solution, is the weighted distance the   ******;
%*****   Min or the Max, and, if Max                             ******;
%***** Is the weighted distance less than the critical limit?    ******;
%**********************************************************************;
       yvy = yvec[1:parms]` * vinverse * yvec[1:parms];
       sqryvy = sqrt(yvy);
       put "sqrt ( Y'(V-1)Y ) =" (sqryvy) 8.3;
   quit;
   run;
%mend bound;

   *******************************************************************;
   ***** after PROC CORR with                                     ****;
   ** var TABL5 TABL10 TABL15 TABL20 TABL30 TABL60 TABL90 TABL120 ****;
   *****  need to select rows, select & rename columns:           ****;
   *******************************************************************;
data pearson; set pearson0(keep=batch _type_ _name_ tabl15 tabl90
                           where=(_name_ in('' 'TABL15' 'TABL90')));
   retain const1 15 const2 90;
   rename tabl15=vn1 tabl90=vn2;
run;
filename ofile 'f:\mydocu~1\shorterm\sesug\connolly\dimen2.txt';
%bound(number=2);

   *******************************************************************;
   ***** after PROC CORR with                                     ****;
   ** var TABL5 TABL10 TABL15 TABL20 TABL30 TABL60 TABL90 TABL120 ****;
   *****  need to rename columns:                                 ****;
   *******************************************************************;
data pearson; set pearson0;
   retain const1 5 const2 10 const3 15 const4 20 const5 30
          const6 60 const7 90 const8 120;
   rename tabl5=vn1 tabl10=vn2 tabl15=vn3 tabl20=vn4 tabl30=vn5
          tabl60=vn6 tabl90=vn7 tabl120=vn8;
run;
filename ofile 'f:\mydocu~1\shorterm\sesug\connolly\dimen8.txt';
%bound(number=8);
```