

Supporting Information

An overall uncertainty approach for the validation of analytical separation methods

T. Saffaj^{*1,2}, B. Ihssane², F. Jhilal³, H. Bouchafra², S. Laslami², S. Alaoui Sosse²

1 : Centre Universitaire Régional d'Interface

Université Sidi Mohamed Ben Abdallah BP 2626 route d'Immouzer-Fès, Morocco.

E-mail: saffajt@gmail.com , taoufiq.saffaj@usmba.ac.ma

2 : Laboratoire de Chimie Organique Appliquée

Faculté des Sciences et Techniques BP 2202 route d'Immouzer-Fès, Morocco

3 : Laboratoire d'Ingénierie des Matériaux, de Modélisation et d'Environnement

Faculté des Sciences Dhar El Mahraz BP 1796 Atlas-Fès, Morocco

Supplement 1: Design for method validation

During the validation of analytical methods, the experimental design involves to make measurements in different days, with replicate determinations within each day. A statistical model to describe the measured values is given by:

$$Y_{ij} = \mu + b_i + e_{ij}; \quad j = 1, 2, \dots, n; i = 1, 2, \dots, a \quad (1)$$

Where Y_{ij} denote the j th replicate observation corresponding to i th day, μ is an unknown general mean, b_i 's represent random day effect for the i th day. e_{ij} 's represent random error under repeatability conditions for the i th day and the j th replicate. It is assumed that b_i 's and e_{ij} 's are all independent having the distributions $b_i \sim N(0, \sigma_b^2)$ and $e_{ij} \sim N(0, \sigma_e^2)$. Thus, $Y_{ij} \sim N(0, \sigma_b^2 + \sigma_e^2)$ and σ_b^2 and σ_e^2 represent the two variance components in the model.

We define

$$\bar{Y} = \frac{1}{an} \sum_{i=1}^a \sum_{j=1}^n Y_{ij}, \bar{Y}_i = \frac{1}{n} \sum_{j=1}^n Y_{ij}, SS_b = n \sum_{i=1}^a (\bar{Y}_i - \bar{Y})^2 \text{ and } SS_e = \sum_{i=1}^a \sum_{j=1}^n (Y_{ij} - \bar{Y}_i)^2, \quad (2)$$

Also we identify the expected mean squares under the model (1):

$$MS_b = \frac{SS_b}{a-1} \text{ and } MS_e = \frac{SS_e}{a(n-1)} \quad (3)$$

The ANOVA estimators of σ_b^2 and σ_e^2 are given by

$$\hat{\sigma}_b^2 = \frac{1}{n} (MS_b - MS_e) \text{ and } \hat{\sigma}_e^2 = MS_e \quad (4)$$

The two-sided tolerance limits that we shall construct will be functions of \bar{Y} , $\hat{\sigma}_b^2$ and $\hat{\sigma}_e^2$. It will be as follows:

$$\bar{Y} \pm k \hat{\sigma}_m \quad (5)$$

Where:

$$\hat{\sigma}_m^2 = \hat{\sigma}_b^2 + \hat{\sigma}_e^2 \quad (6)$$

Supplement 2: Interpretation of uncertainty profile

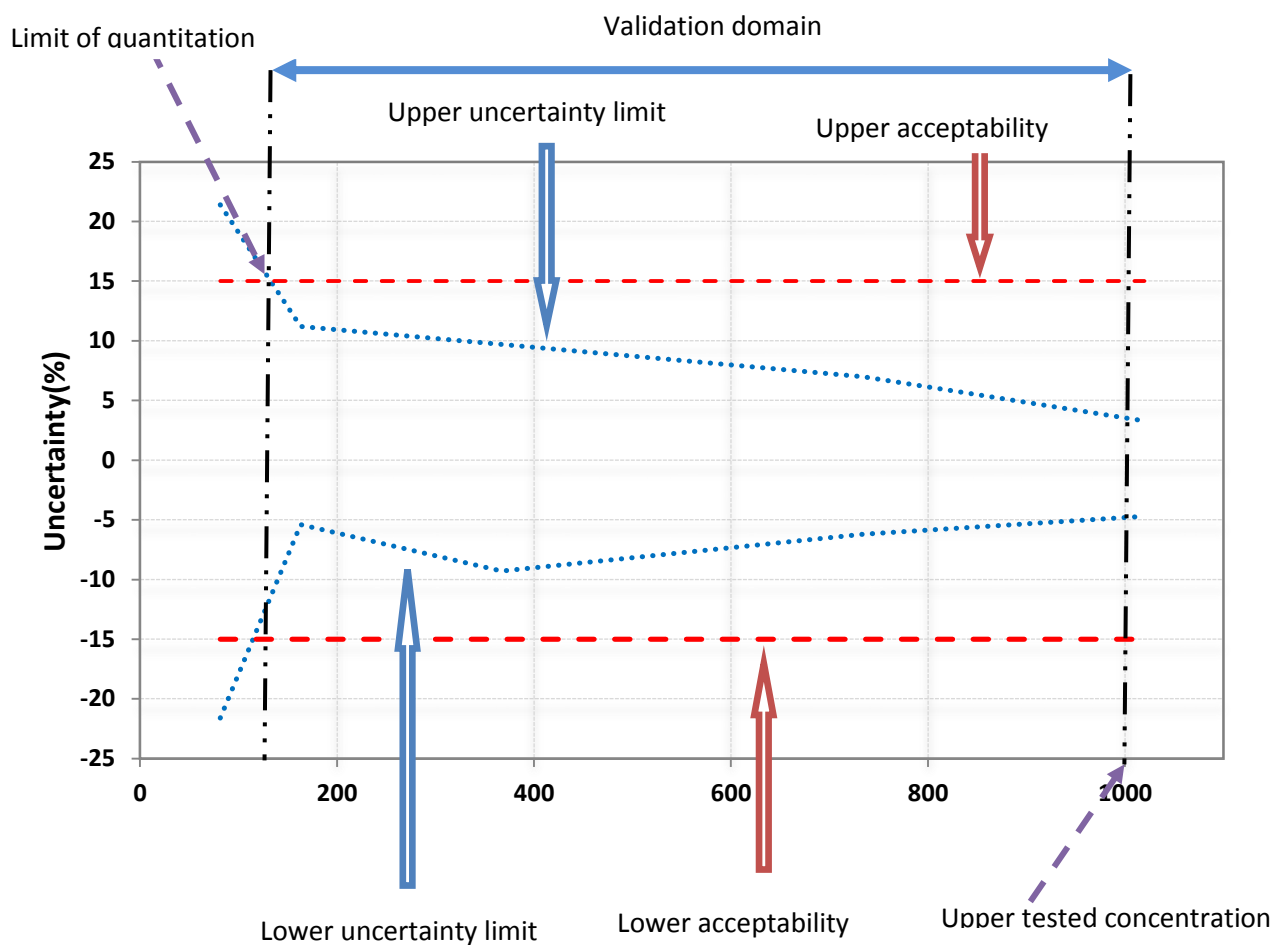


Fig. S2: Schematic representation of an uncertainty profile.

Uncertainty profile proposes a graphical method for establishing whether the combination of precision and trueness of a given method is correctly adapted to the analytical problem within a requested analytical range. To obtain an uncertainty profile, the relative uncertainty is plotted versus the validation standards concentration levels and the upper uncertainty limits are jointed together on one hand and the lower limits on the other hand. Finally, the pre-specified acceptance limits are reported on the graph.

A method is claimed to be valid when uncertainty intervals are fully included within the acceptability limits. A validity domain is, then, defined between the limit of quantitation and the upper tested concentration

Program S3. Matlab program to implement the calculation of two-sided β -content, γ -confidence tolerance intervals and to estimate the uncertainty of measurement using Mee's method.

```
function yf = Meegamma(X,gamma,beta,Xr)
%Meegamma is function for tolerance interval computation
% yf = [Yl;Yu] gives the predicted values of the
% beta gamma tolerance interval (Mee method)and estimate the
% uncertainty of measurement as well as calculate uncertainty limits.
% X: data matrix contains values of observed variables, data ordered by
days
% Xr: true value
% beta: required coverage of the interval(for example beta=0.667)
% gamma: required guarantee of the interval (for example gamma=0.90)

[n,a] = size(X); % n: number of replicates per series/day; a: number of
series/day
[d,table] = anova1(X);
figure(1);
close 1
figure(2);
close 2

Ymoy = mean(mean(X));
MSb = cell2mat(table(2,4));
MSe = cell2mat(table(3,4));
sigma2t = max(0,(1/n)*(MSb-MSe));
sigma2e = MSe;

if gamma==0.9;
    w=0.15;
elseif gamma==0.95;
    w=0.095;
    elseif gamma==0.99
        w=0.025;
end
F = icdf('f',w,a-1,a*(n-1));
R = max(0,(1/n)*(MSb/(F*MSe)-1));
R0 = (R+1)/((n*R)+1);
f = (R+1)^2 /(((R+(1/n))^2 / (a-1))+((1-(1/n))/(a*n)));
m = icdf('chi2',1-gamma,f);
mn = icdf('ncx2',beta,1,1/(a*n*R0));
Km = (f*mn/m)^0.5;

R1 = max(0,(1/n)*((MSb/MSe)-1));
f1 = (R1+1)^2 /(((R1+(1/n))^2 / (a-1))+((1-(1/n))/(a*n)));
t = icdf('t',(1+gamma)/2,f1);

U = Ymoy + Km*(sigma2t+sigma2e)^0.5; %upper tolerance interval
L = Ymoy - Km*(sigma2t+sigma2e)^0.5; % lower tolerance interval

U100 = ((U-Xr)/Xr)*100; % relative upper tolerance interval
L100 = ((L-Xr)/Xr)*100; % relative lower tolerance interval

UN = (U-L)/(2*t); % uncertainty of measurement
UN100 = (UN/Xr)*100; % relative uncertainty of measurement
```

```
UEXP= 2*UN; % expanded uncertainty of measurement
UEXP100 = (UEXP/Xr)*100; % relative expanded uncertainty of measurement

u100 = ((Ymoy-Xr+UEXP)/Xr)*100; % relative upper uncertainty limit
l100 = ((Ymoy-Xr-UEXP)/Xr)*100; % relative lower uncertainty limit

uabs = Ymoy+UEXP; % absolute upper uncertainty limit
labs = Ymoy-UEXP; % absolute lower uncertainty limit

yf = [L,U;L100,U100;UEXP,UEXP100;labs,uabs;l100,u100];
```

Program S4. Matlab program to implement the calculation of two-sided β -content, γ -confidence tolerance intervals as well as to calculate uncertainty limits using Liao-Lin-Iyer approach.

```
function yf = Liaogamma(N,X,gamma,beta,Xr)
%Liaogamma is function for tolerance interval computation
% yf = [Yl;Yu] gives the predicted values of the beta gamma tolerance
interval using LIA METHOD and estimate the
% uncertainty of measurement as well as calculate uncertainty limits
% N : number of simulation
% X: data matrix contains values of observed variables, data ordered by
days
% Xr: true value
% beta: required coverage of the interval(for example beta=0.667)
% gamma: required guarantee of the interval (for example gamma=0.90)

[n,a] = size(X);
[d,table] = anova1(X);
SSb = cell2mat(table(2,2));
SSe = cell2mat(table(3,2));
figure(1);
close 1
figure(2);
close 2

Ymoy = mean(mean(X));
Ut2i = chi2rnd(a-1,N,1);
Ue2i = chi2rnd(a*(n-1),N,1);
Zi = icdf('Normal',(1+beta)/2,0,1);
D = (((1/n)*(1+(1/a))*SSb./Ut2i)+((1-(1/n))*SSe./Ue2i));
gamma1= gamma*100
D1 = prctile(D,gamma1);
D2 = Zi*D1^0.5;

Yl = Ymoy-D2; % lower tolerance interval
Yu = Ymoy+D2; % upper tolerance interval

U100 = ((Yu-Xr)/Xr)*100; % relative upper tolerance interval
L100 = ((Yl-Xr)/Xr)*100; % relative lower tolerance interval

MSb = cell2mat(table(2,4));
MSe = cell2mat(table(3,4));
R1 = max(0,(1/n)*((MSb/MSe)-1));
f1 = (R1+1)^2 / (((R1+(1/n))^2 / (a-1))+((1-(1/n))/(a*n)));
t = icdf('t',(1+gamma)/2,f1);

UN = (Yu-Yl)/(2*t); % uncertainty of measurement
UEXP= 2*UN; % expanded uncertainty of measurement
UEXP100 = (UEXP/Xr)*100; % relative expanded uncertainty of measurement

u100 =((Ymoy-Xr+UEXP)/Xr)*100; % relative upper uncertainty limit
l100 =((Ymoy-Xr-UEXP)/Xr)*100; % relative lower uncertainty limit

uabs = Ymoy+UEXP; % absolute upper uncertainty limit
labs = Ymoy-UEXP; % absolute lower uncertainty limit

yf = [Yl,Yu;L100,U100;UEXP,UEXP100;labs,uabs;l100,u100];
```

Program S5. Matlab program, to implement the calculation of two-sided β -content, γ -confidence tolerance intervals as well as to estimate the uncertainty limits using MLS procedure.

```
function yf = MLsgamma(X, gamma, beta, Xr)
%MLsgamma is function for tolerance interval computation
% yf = [Yl;Yu] gives the predicted values of the
% beta gamma tolerance interval WITH MLS METHODOLOGY and estimate the
% uncertainty of measurement
% X: data matrix contains values of observed variables, data ordered by
days
% Xr: true value
% beta: required coverage of the interval (for example beta=0.667)
% gamma: required guarantee of the interval (for example gamma=0.90)

[n,a] = size(X);
[d,table] = anova1(X);
SSb = cell2mat(table(2,2));
SSe = cell2mat(table(3,2));
Ymoy = mean(mean(X));
Ut2 = icdf('chi2', (1-gamma), a-1);
Ue2 = icdf('chi2', (1-gamma), a*(n-1));
Zi = icdf('Normal', (1+beta)/2, 0, 1);
D = (((1/n)*(1+(1/a))*SSb/(a-1))+((1-(1/n))*SSe/(a*(n-1))))+(((1/n)*(1+(1/a))*SSb/(a-1))^2*((a-1-Ut2)/Ut2)^2+((1-(1/n))*SSe/(a*(n-1)))^2*((a*(n-1)-Ue2)/Ue2)^2)^0.5;
D2 = Zi*D^0.5;

Yl = Ymoy-D2; % lower tolerance interval
Yu = Ymoy+D2; % upper tolerance interval

U100 = ((Yu-Xr)/Xr)*100; % relative upper tolerance interval
L100 = ((Yl-Xr)/Xr)*100; % relative lower tolerance interval

figure(1);
close 1
figure(2);
close 2

MSb = cell2mat(table(2,4));
MSe = cell2mat(table(3,4));
R1 = max(0, (1/n)*((MSb/MSe)-1));
f1 = (R1+1)^2 / (((R1+(1/n))^2 / (a-1))+((1-(1/n))/(a*n)));
t = icdf('t', (1+gamma)/2, f1);

UN = (Yu-Yl)/(2*t);
UEXP= 2*UN;
UEXP100 = (UEXP/Xr)*100;

u100 = ((Ymoy-Xr+UEXP)/Xr)*100; % relative upper uncertainty limit
l100 = ((Ymoy-Xr-UEXP)/Xr)*100; % relative lower uncertainty limit

uabs = Ymoy+UEXP; % absolute upper uncertainty limit
labs = Ymoy-UEXP; % absolute lower uncertainty limit

yf = [Yl, Yu; L100, U100; UEXP, UEXP100; labs, uabs; l100, u100];
```

Supplement 6: example of data for illustration of the calculations

Table S1: calculated concentrations for the first level of concentration of *HPAE-PAD* method for the determination of 1-kestose in fruit juices. Concentration expressed as mg 100mL⁻¹

Replicate	Theoretical concentration	Day		
		1	2	3
1	0.25	0.233	0.233	0.232
2		0.236	0.238	0.243
3		0.244	0.242	0.239