

Plate Reader 1

Step 1: data loading and preparation

Add PLATERO set of functions to your working directory:

```
my = version('-release');
if str2double(my(1:4))<2020
addpath(genpath('rprev2020'))
else
addpath(genpath('r2020'))
end
```

Now, load the data resulting from the calibration experiment, written in "filename". This data is organized by sheets, where each sheet has one repetition of the measurements.

```
filename = "PlateReader1.xlsx";
colnames = {'WellID','Well','Concentration','G50','G60','G70','G80','OD'};
[dataPR, indgfp] = readexperiment(filename,"B102:I197",50:10:80,false,colnames);
size(dataPR)
```

ans = 1x2
3072 6

Divide the dataset into the subset with medium values (dataPRblk) and the set with fluorescein values (dataPRgfp).

```
datPRblk = dataPR(~indgfp,:);
datPRgfp = dataPR(indgfp,:);
disp(strcat("This data set has ", string(size(datPRblk,1)), ...
" BLK observations and ", string(size(datPRgfp,1)), ...
" GFP observations."))
```

This data set has 512 BLK observations and 2560 GFP observations.

Obtain the partition of the fluorescein dataset into the model building set (70%) and the model validation set (30%). A seed is set to ensure reproducibility of the results. The resulting subsets are stored as the `calibration_PR1.mat` and the `validation_PR1.mat` files.

```
rng(0207)
[datagfp_cal, datagfp_val] = cvsplit(datPRgfp, 0.7);
disp(strcat("The calibration data set has ", string(size(datagfp_cal,1)), ...
" observations and the validation data set has ", ...
string(size(datagfp_val,1)), " observations."))
```

The calibration data set has 1760 observations and the validation data set has 800 observations.

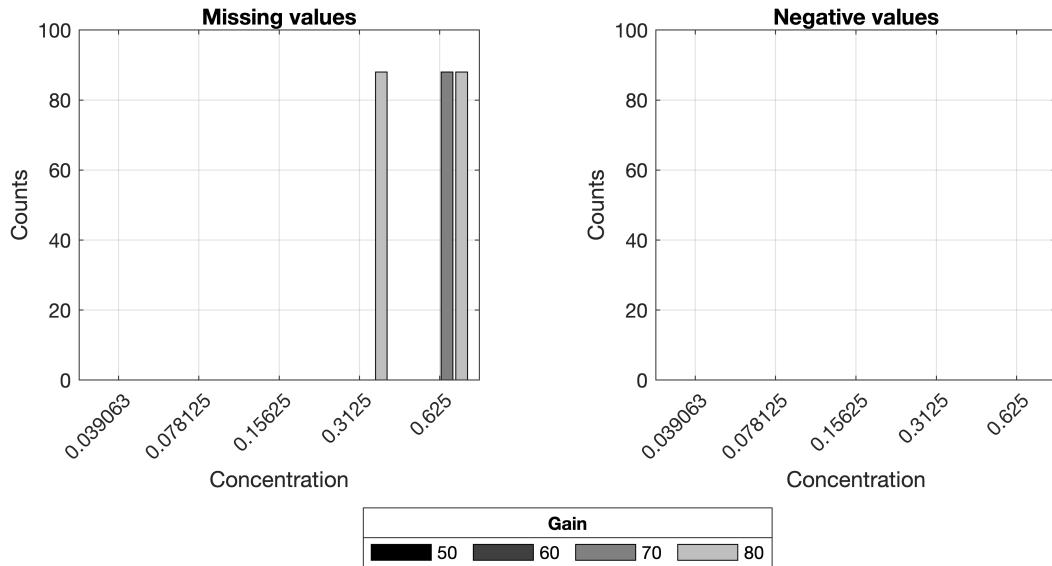
```
data_cal_pr1 = [datPRblk; datagfp_cal];
save('calibration_PR1.mat','data_cal_pr1')
save('validation_PR1.mat','datagfp_val')
```

Step 2: Model Building step

Load the calibration subset, fit the model and store the coefficients.

```
data_cal_pr1 = load("calibration_PR1.mat").data_cal_pr1;  
[blk_data, flu_data_PR1] = explore_data(data_cal_pr1, nan);
```

Explorative plot of missing data for each concentration level:



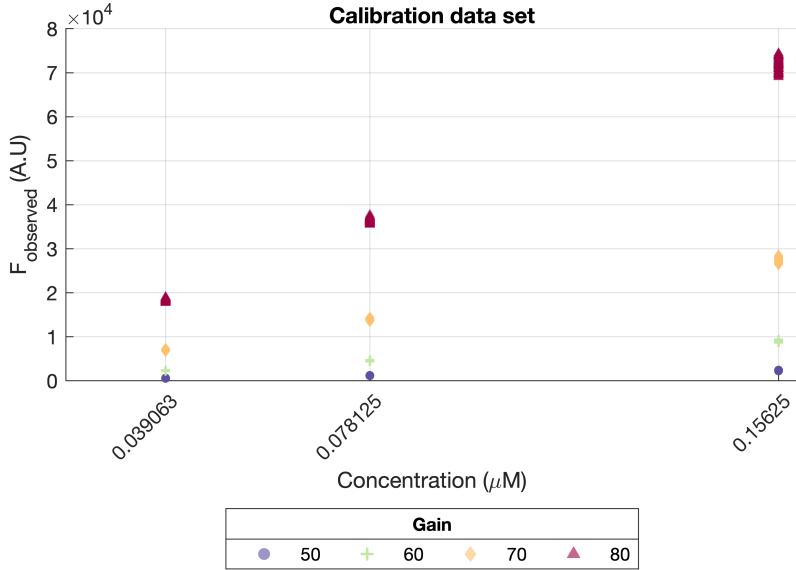
Concentration(s) 0.3125, 0.625 had missing values

Concentration(s) had negative values

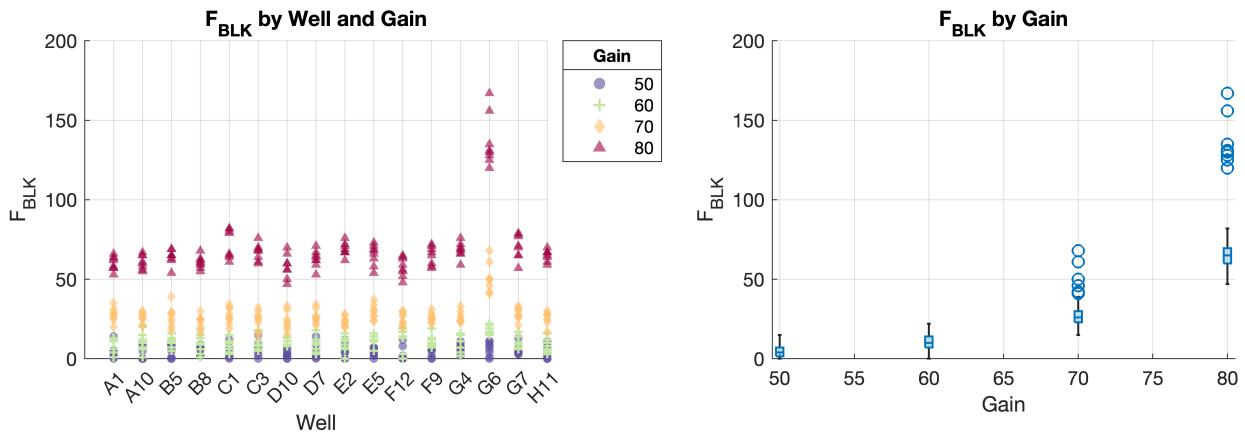
Gain(s) 70, 80 had missing values

Gain(s) had negative values

Explorative plot of the raw F_observed Fluorescein data:



Explorative plot of the raw F_BLK data:



```
[flu_data_PR1, modelPR1, calmetrics_PR1] = fit_plattero_model(blk_data, flu_data_PR1, ...)
```

Fit f_G and plot corrected data (F_reporter):
ANOVA on coefficient b_1 for all levels of concentration

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Concentration	0	2	1.1491e-06	0.13	0.874
Well(Concentration)	0.00035	30	1.16589e-05	1.37	0.1049
Error	0.00197	231	8.52553e-06		
Total	0.00232	263			

Constrained (Type III) sums of squares.

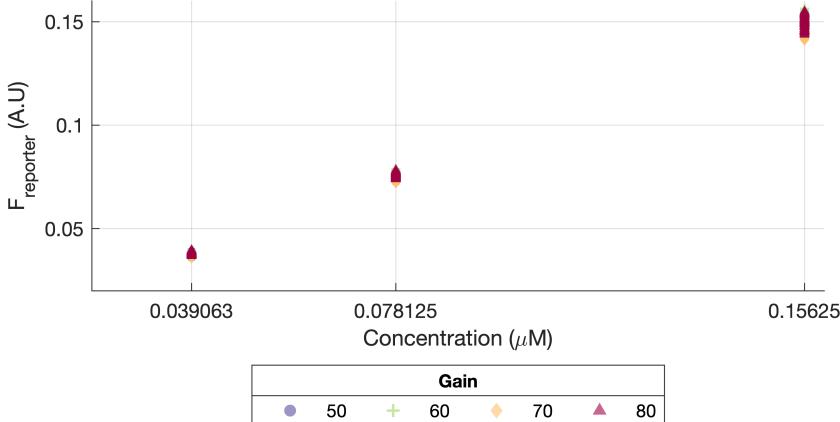
ANOVA on coefficient b_2 for all levels of concentration

Analysis of Variance

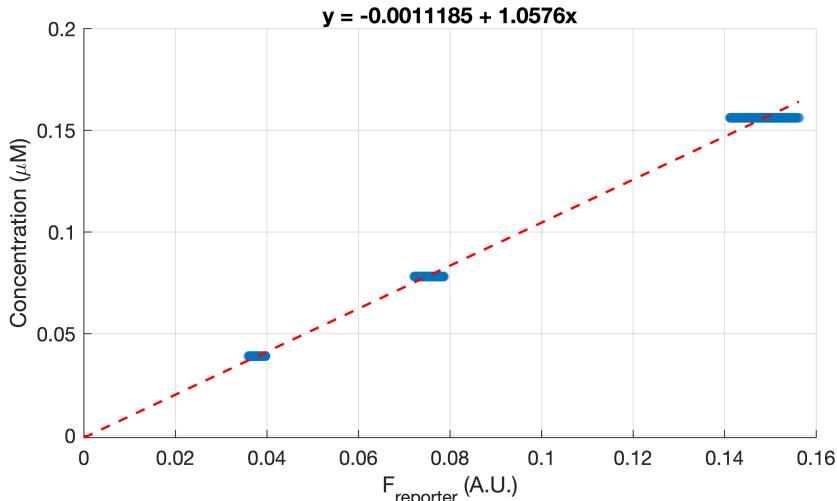
Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Concentration	2.19985e-10	2	1.09993e-10	0.23	0.795
Well(Concentration)	1.92842e-08	30	6.42808e-10	1.34	0.1191
Error	1.10662e-07	231	4.79055e-10		
Total	1.30166e-07	263			

Constrained (Type III) sums of squares.

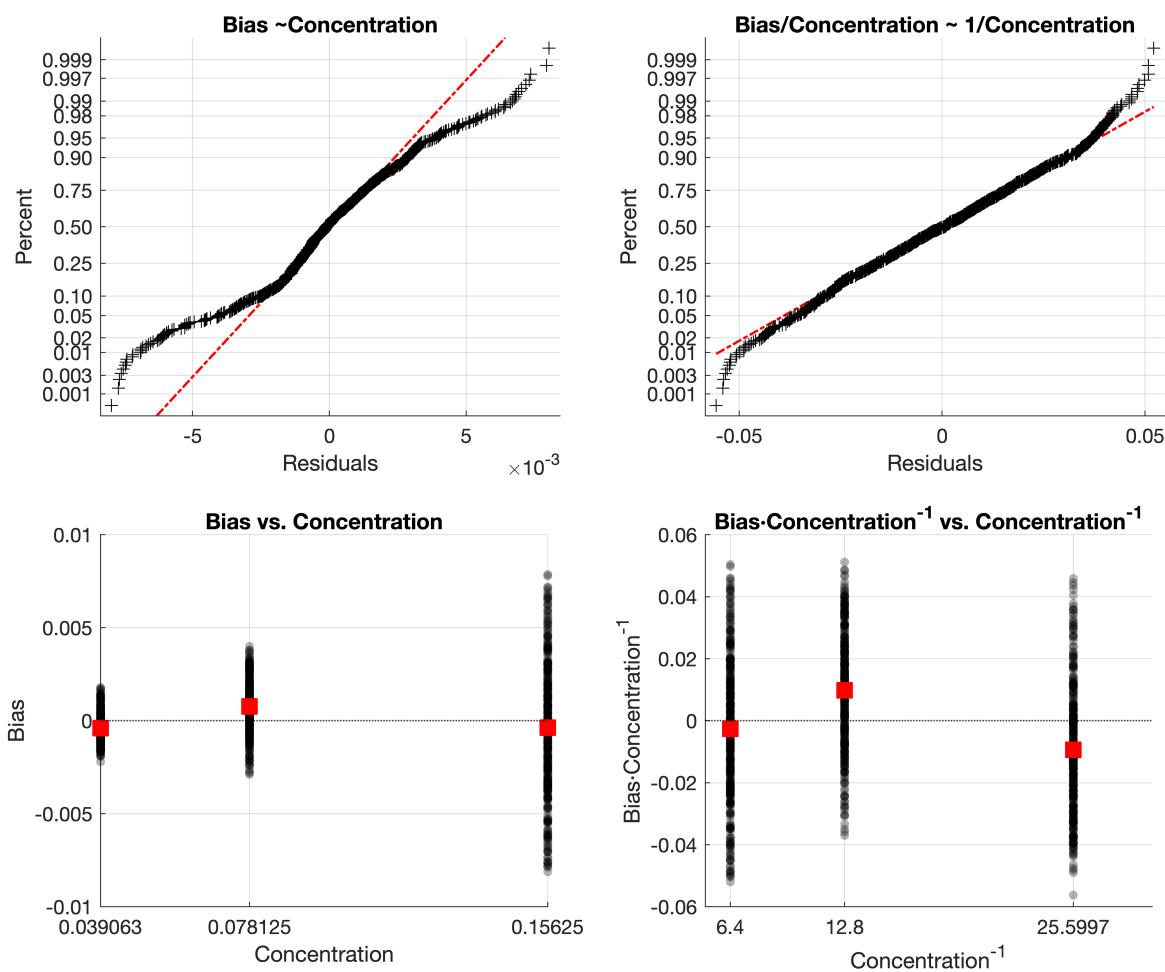
Data processed with gain (exp. model) correction

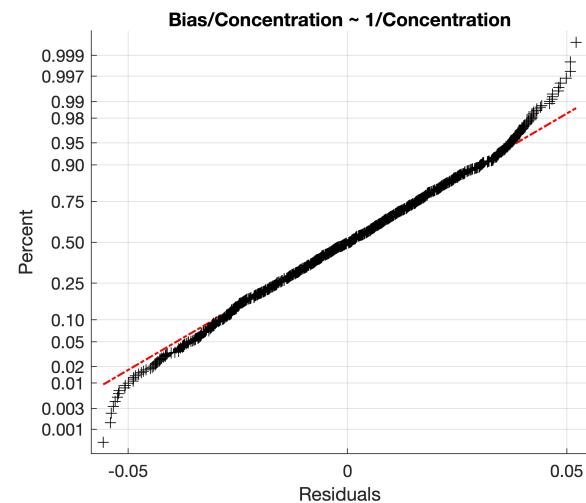
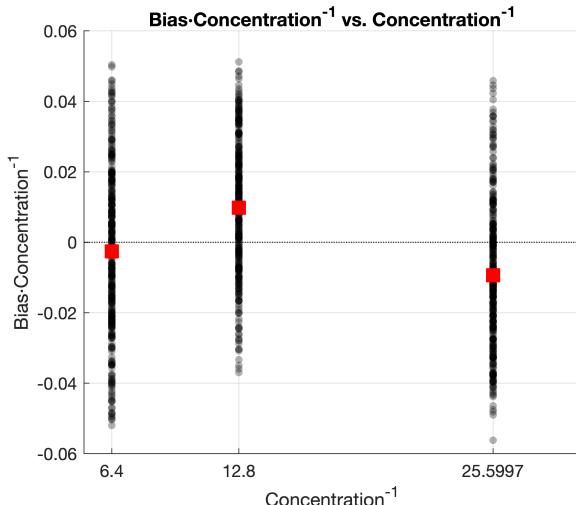


Fit f_UC and plot estimated concentration data (C):



Analyze the bias in the predictions and estimate the uncertainty in the predictions (s_{Bias}):





Compute error metrics for the Model Building step:

F_BLK (G = 50)

F_BLK (G = 60)

F_BLK (G = 70)

F_BLK (G = 80)

b1

b2

4

10

26

65

0.24298

-0.0009933

-0.00

Step 3: Model Validation step

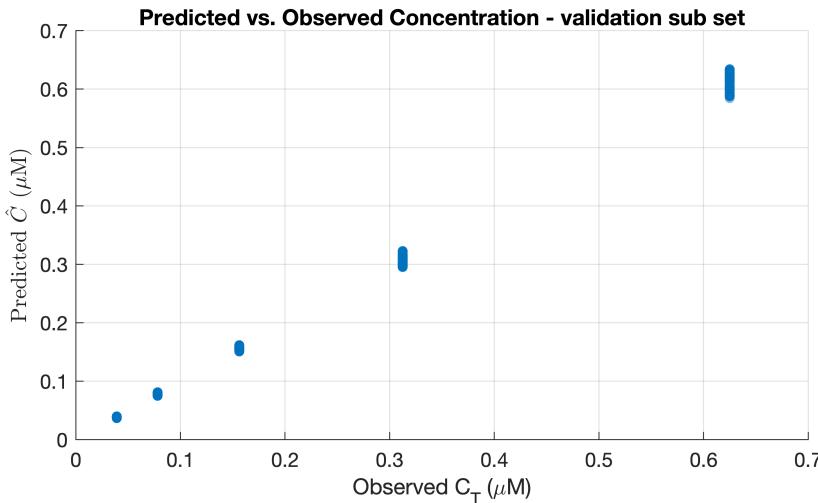
```
datagfp_val = load("validation_PR1.mat").datagfp_val;
uG = unique(flu_data_PR1.Gain);
uC = unique(flu_data_PR1.Concentration);
data_val_pr1 = datagfp_val(ismember(datagfp_val.Gain, uG), :);
G = unique(data_val_pr1.Gain);

% Assign the corresponding F_BLK values to each observation F_obs
data_val_pr1.Fblk = repmat(modelPR1{:,1:4}', size(data_val_pr1,1)/length(G),1);

% Run the model on the validation set
[data_val_pr1, valmetrics_inrange, vprocv] = use_platero_model(data_val_pr1, ...
    modelPR1,"PR_1");
```

A 15 % of the observations was missing.

Plot the Validation dataset transformed to concentration units:



R&R Analysis:

R & R Analysis on measurements for $C = 0.039063$

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Reprod (Gain), $C = (0.039063)$	2.59442e-05	3	8.64806e-06	124.88	8.06005e-41
Replicates, $C = (0.039063)$	3.67816e-05	4	9.19539e-06	132.79	1.49776e-48
Error	1.05257e-05	152	6.92483e-08		
Total	7.32515e-05	159			

[Constrained \(Type III\) sums of squares.](#)

R & R Analysis on measurements for $C = 0.078125$

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Reprod (Gain), $C = (0.078125)$	0.00012	3	4.01839e-05	288.55	1.59411e-62
Replicates, $C = (0.078125)$	0.0002	4	4.94217e-05	354.89	5.51902e-76
Error	0.00002	152	1.39259e-07		
Total	0.00034	159			

[Constrained \(Type III\) sums of squares.](#)

R & R Analysis on measurements for $C = 0.15625$

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Reprod (Gain), $C = (0.15625)$	0.00041	3	0.00014	266.68	2.51245e-60
Replicates, $C = (0.15625)$	0.00059	4	0.00015	287.71	8.34823e-70
Error	0.00008	152	0		
Total	0.00107	159			

[Constrained \(Type III\) sums of squares.](#)

R & R Analysis on measurements for $C = 0.3125$

Analysis of Variance

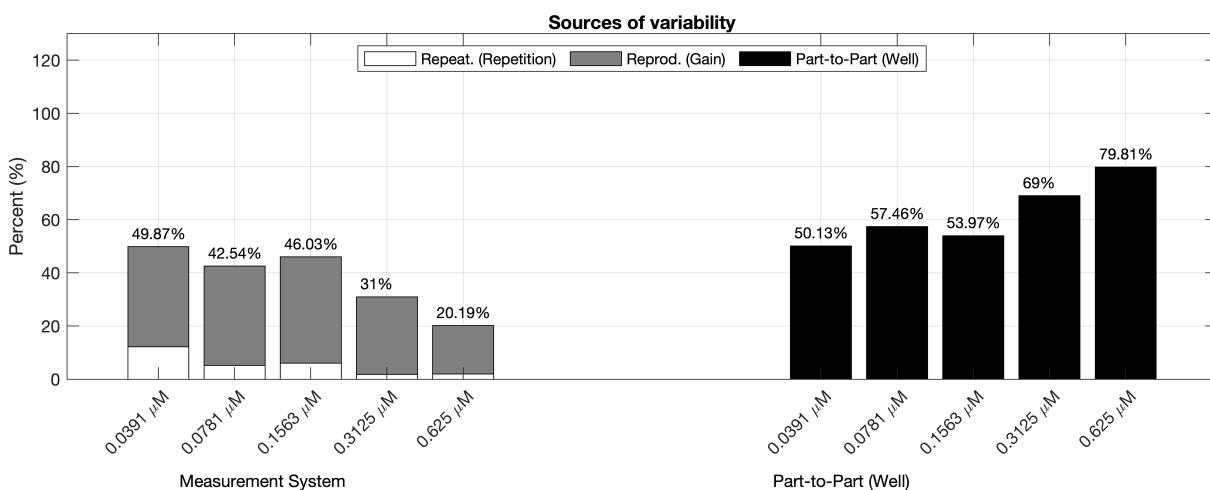
Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Reprod (Gain), $C = (0.3125)$	0.00152	2	0.00076	655.78	6.5467e-63
Replicates, $C = (0.3125)$	0.0043	4	0.00108	929.06	1.99528e-85
Error	0.00013	113	0		
Total	0.00595	119			

[Constrained \(Type III\) sums of squares.](#)

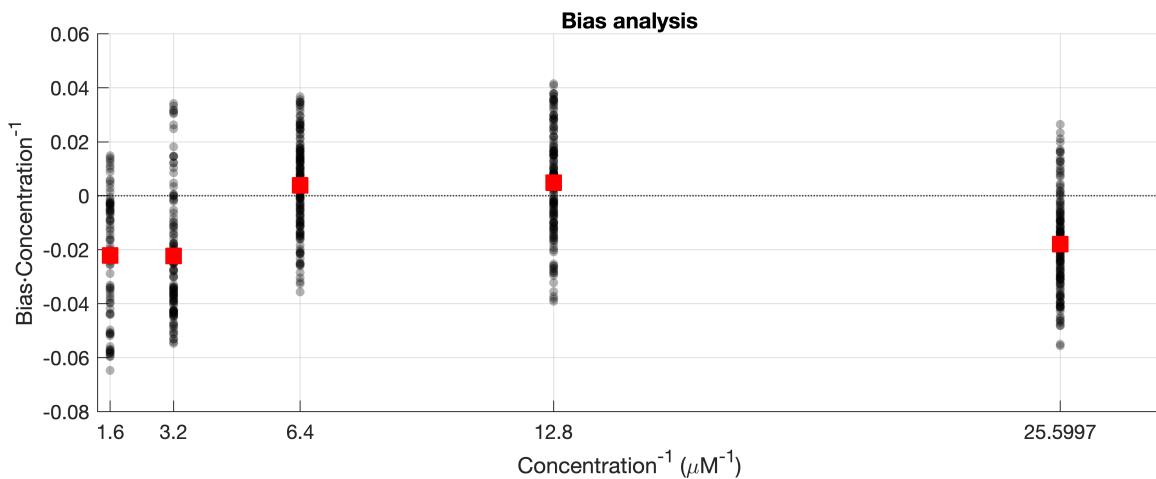
R & R Analysis on measurements for $C = 0.625$

Analysis of Variance					
Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Reprod (Gain), C = (0.625)	0.00202	1	0.00202	372.82	1.29067e-30
Replicates, C = (0.625)	0.01411	4	0.00353	652.13	7.47094e-57
Error	0.0004	74	0.00001		
Total	0.01653	79			

Constrained (Type III) sums of squares.



B&L Analysis:



Linear regression model:

$$y \sim 1 + x_1$$

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	-0.0078207	0.0014307	-5.4665	6.4573e-08
x1	-7.6231e-05	0.00010005	-0.76196	0.44635

Number of observations: 680, Error degrees of freedom: 678

Root Mean Squared Error: 0.0229

R-squared: 0.000856, Adjusted R-Squared: -0.000618

F-statistic vs. constant model: 0.581, p-value = 0.446

Contribution of model terms to the total bias variability:

Bias Model - linear term (%): 0.7821 %

Bias Model - bias term (%): 10.1993 %

Confidence Intervals and Error metrics:

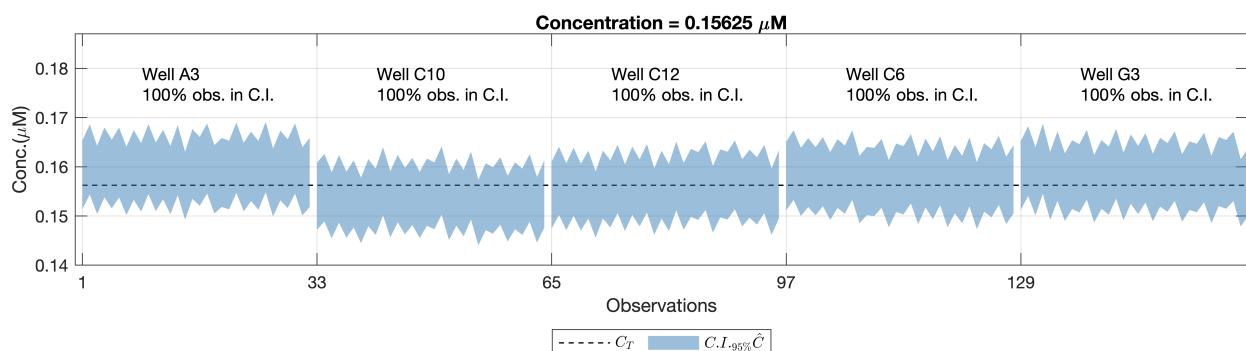
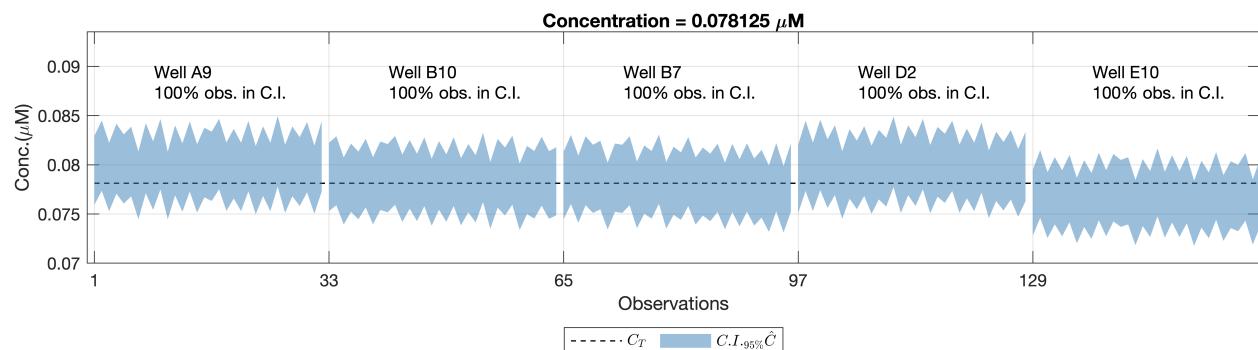
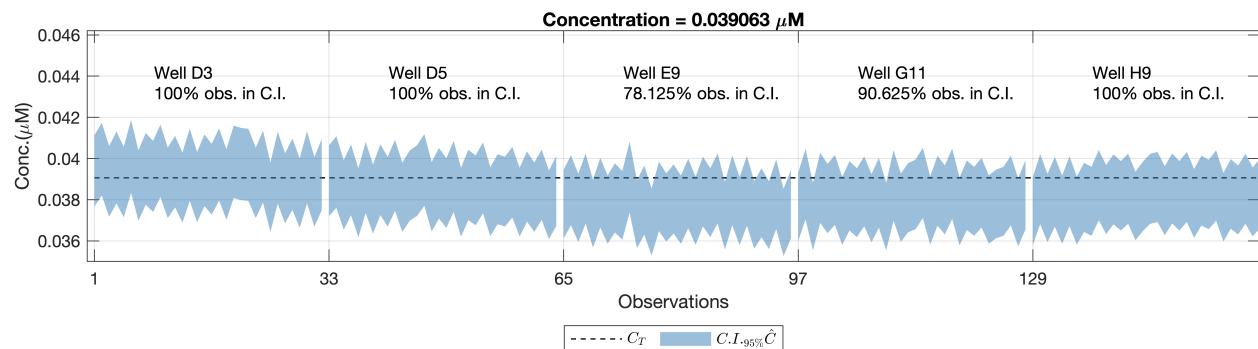
pctgeci: 91.6176

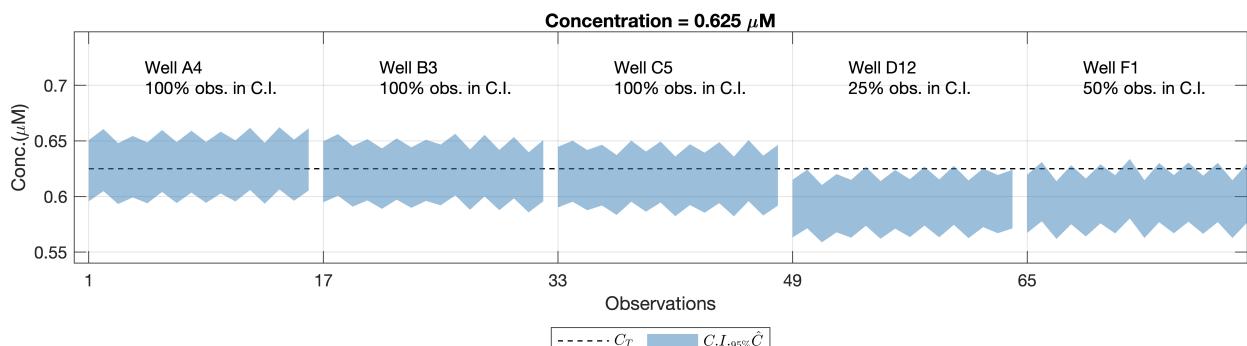
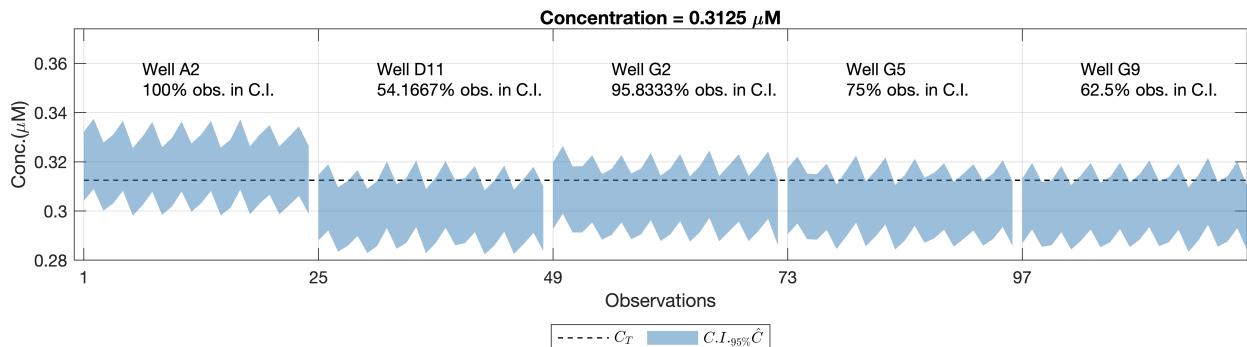
mse: 6.6503e-05

relerr: [680x1 double]

minrelerror: 1.7022e-05

maxrelerror: 0.0647





```
% Comparison between calibration-set and validation-set metrics
% load(strcat(dirdatasave,'cal_results.mat'))
perftable = table([calmetrics_PR1.mse;valmetrics_inrange.mse],...
[calmetrics_PR1.minrelerror;valmetrics_inrange.minrelerror]*100,... 
[calmetrics_PR1.maxrelerror;valmetrics_inrange.maxrelerror]*100,... 
'RowNames',{'Calibration', 'Validation (within range)'},...
'VariableNames',{'MSE','Min.Rel.Error (%)','Max.Rel.Error (%)'});
display(perftable)
```

perftable = 2x3 table

	MSE	Min.Rel.Error (%)	Max.Rel.Error (%)
1 Calibration	5.6728e-06	0.0040	5.6197
2 Validation (within range)	6.6503e-05	0.0017	6.4732