

Plate Reader 2 -- experiment 2

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 = "PlateReader2_exp2.xlsx";
colnames = {'WellID','Well','Concentration','G50','G60','G70','G80','G90','G120'};
[dataPR, indgfp] = readexperiment(filename,"A7:I103", [50:10:90,120], false, ...
    colnames, 0);
iG = and(dataPR.Gain > 50, dataPR.Gain < 120);
dataPR = dataPR(iG, :);
indgfp = indgfp(iG, :);
size(dataPR)
```

ans = 1×2
3072 5

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_pr2e2 = [datPRblk; datagfp_cal];
```

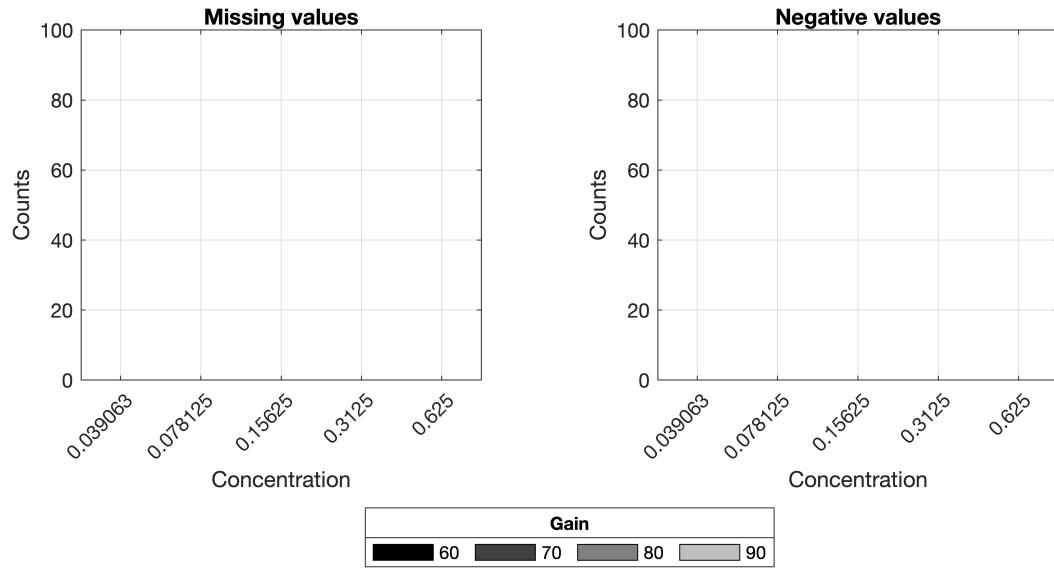
```
save('calibration_PR2e2.mat','data_cal_pr2e2')
save('validation_PR2e2.mat','datagfp_val')
```

Step 2: Model Building step

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

```
data_cal_pr2e2 = load("calibration_PR2e2.mat").data_cal_pr2e2;
[blk_data, flu_data_PR2e2] = explore_data(data_cal_pr2e2, 0);
```

Explorative plot of missing data for each concentration level:



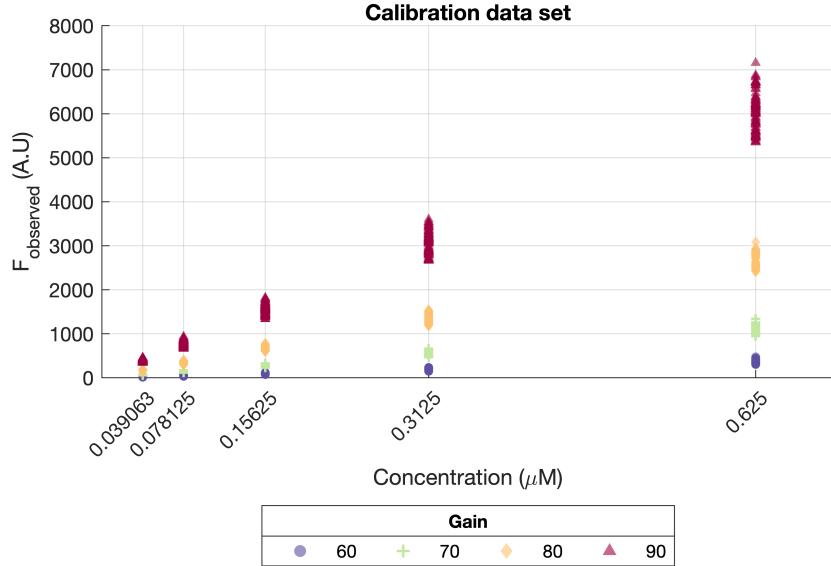
Concentration(s) had missing values

Concentration(s) had negative values

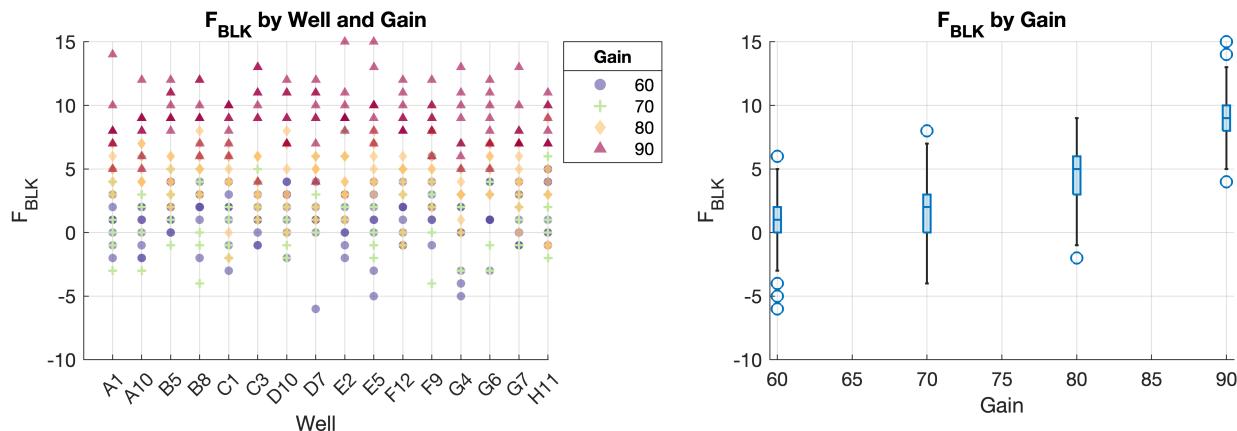
Gain(s) had missing values

Gain(s) had negative values

Explorative plot of the raw F_observed Fluorescein data:



Explorative plot of the raw F_BLK data:



```
[flu_data_PR2e2, modelPR2e2, calmetrics_PR2e2] = fit_platero_model(blk_data, ...
    flu_data_PR2e2, "PR_2e2");
```

Fit f_G and plot corrected data (F_reporter):

ANOVA on coefficient b_1 for all levels of concentration

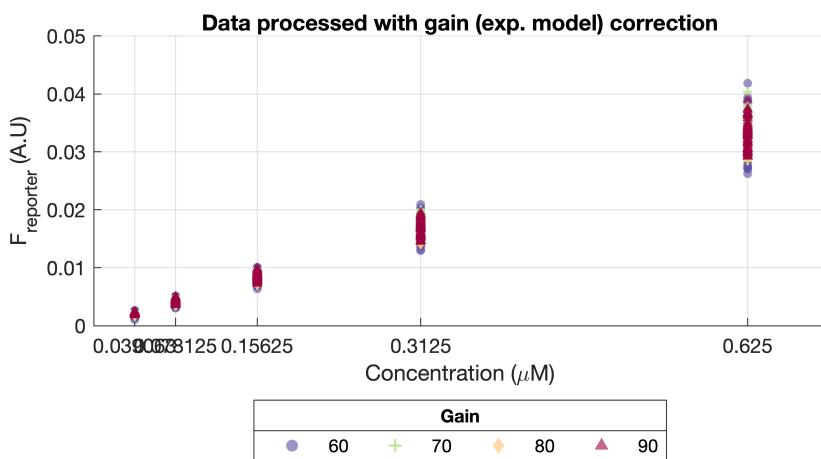
Analysis of Variance					
Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Concentration	0.00978	4	0.00244	2.34	0.0542
Well(Concentration)	0.02453	50	0.00049	0.47	0.9992
Error	0.40145	385	0.00104		
Total	0.43575	439			

Constrained (Type III) sums of squares.

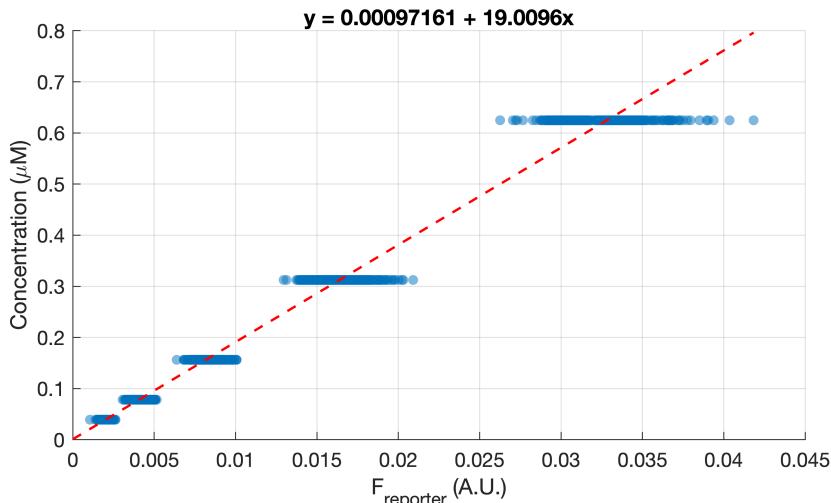
ANOVA on coefficient b_2 for all levels of concentration

Analysis of Variance					
Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Concentration	3.77559e-07	4	9.43897e-08	2.3	0.0584
Well(Concentration)	9.65461e-07	50	1.93092e-08	0.47	0.9992
Error	1.5811e-05	385	4.10676e-08		
Total	1.7154e-05	439			

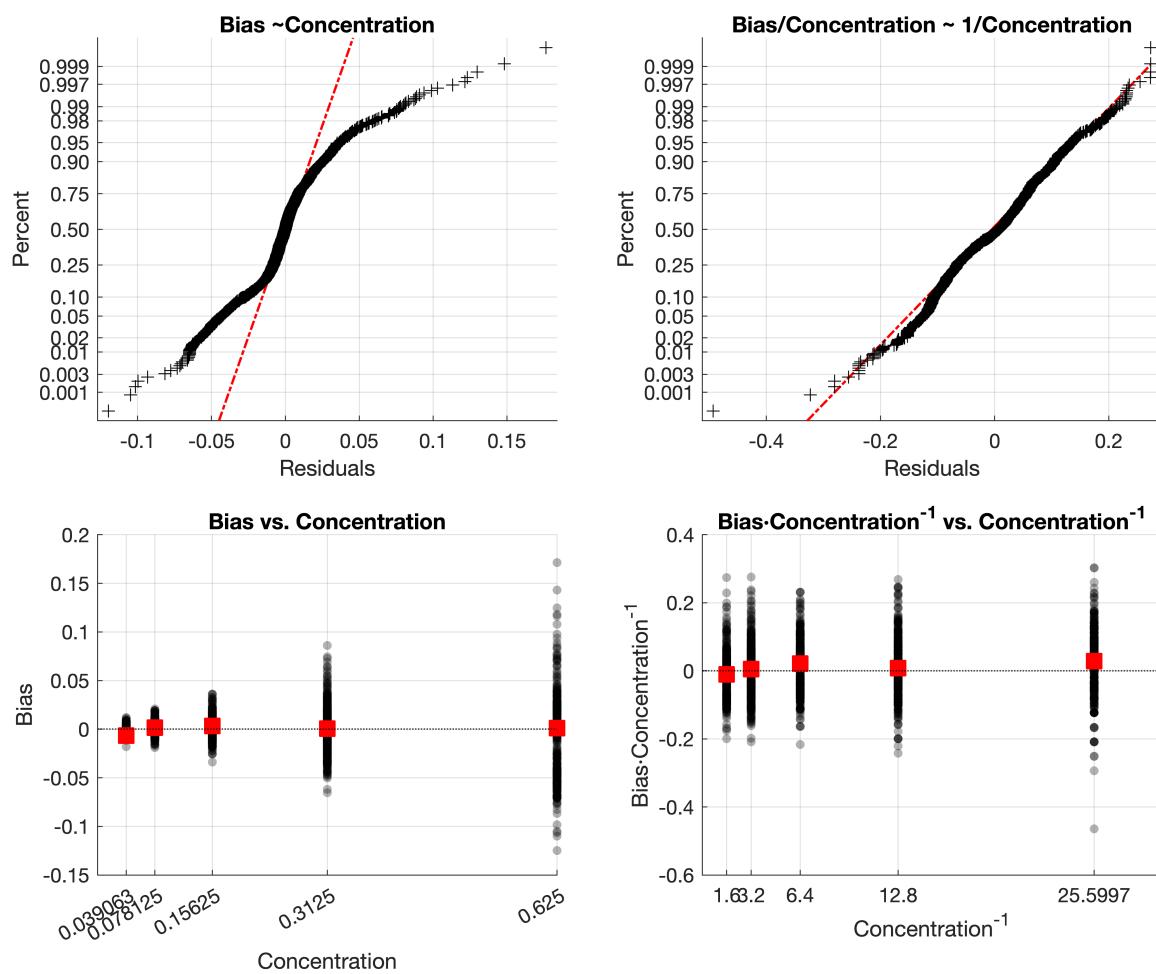
Constrained (Type III) sums of squares.

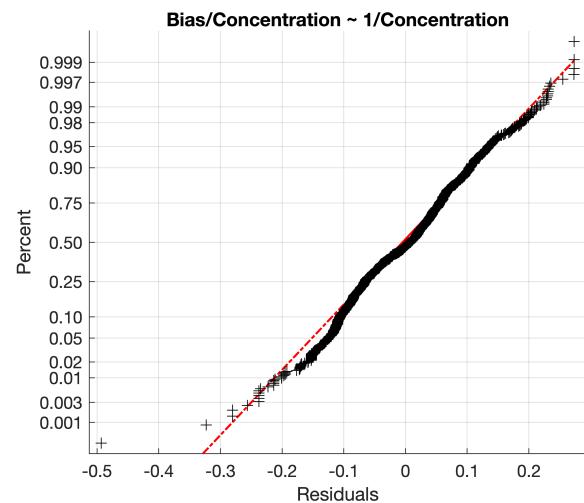
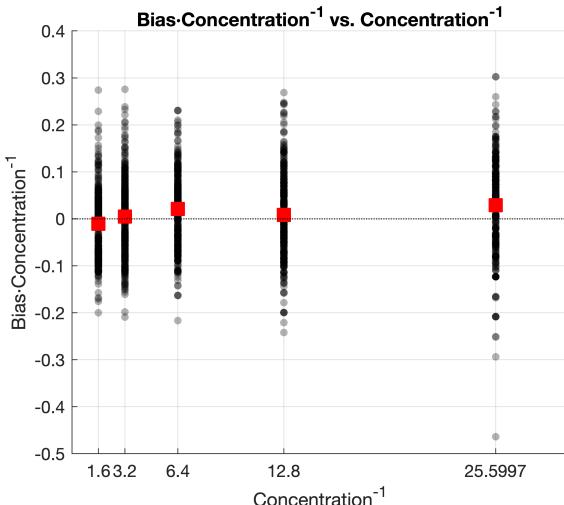


Fit f_UC and plot estimated concentration data (C):



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 = 60)

F_BLK (G = 70)

F_BLK (G = 80)

F_BLK (G = 90)

b1

b2

1

2

5

9

0.19789

-0.00070272

0.00

Step 3: Model Validation step

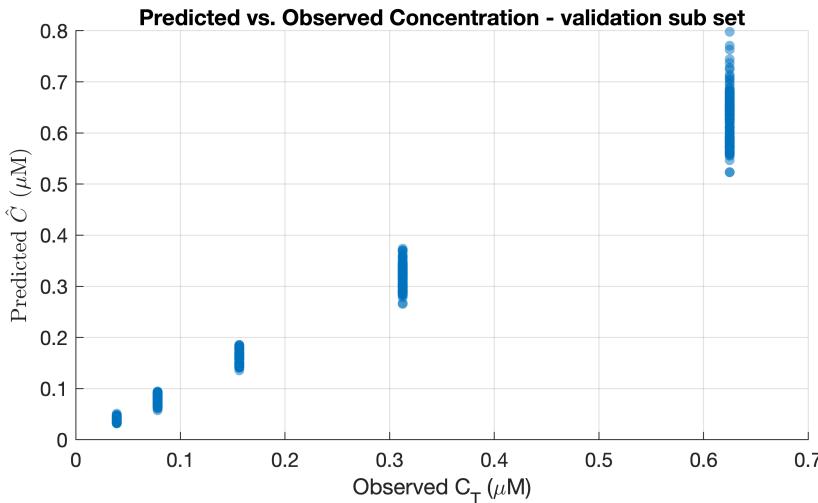
```
datagfp_val = load("validation_PR2e2.mat").datagfp_val;
uG = unique(flu_data_PR2e2.Gain);
uC = unique(flu_data_PR2e2.Concentration);
data_val_pr2e2 = datagfp_val(ismember(datagfp_val.Gain, uG),:);
G = unique(data_val_pr2e2.Gain);

% Assign the corresponding F_BLK values to each observation F_obs
data_val_pr2e2.Fblk = repmat(modelPR2e2{:,1:length(G)}', ...
    size(data_val_pr2e2,1)/length(G),1);

% Run the model on the validation set
[data_val_pr2e2, valmetrics_inrange, vprocv] = use_platero_model(data_val_pr2e2, ...
    modelPR2e2, "PR_2e2");
```

A 0 % 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), $C = (0.039063)$	0.0001	3	0.00003	5.1	0.0022
Replicates, $C = (0.039063)$	0.00094	4	0.00023	34.87	0
Error	0.00102	152	0.00001		
Total	0.00206	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), $C = (0.078125)$	0.00021	3	0.00007	4.12	0.0077
Replicates, $C = (0.078125)$	0.00623	4	0.00156	91.08	0
Error	0.0026	152	0.00002		
Total	0.00904	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), $C = (0.15625)$	0.00076	3	0.00025	5.9	0.0008
Replicates, $C = (0.15625)$	0.00898	4	0.00225	52.07	0
Error	0.00656	152	0.00004		
Total	0.0163	159			

Constrained (Type III) sums of squares.

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

Analysis of Variance

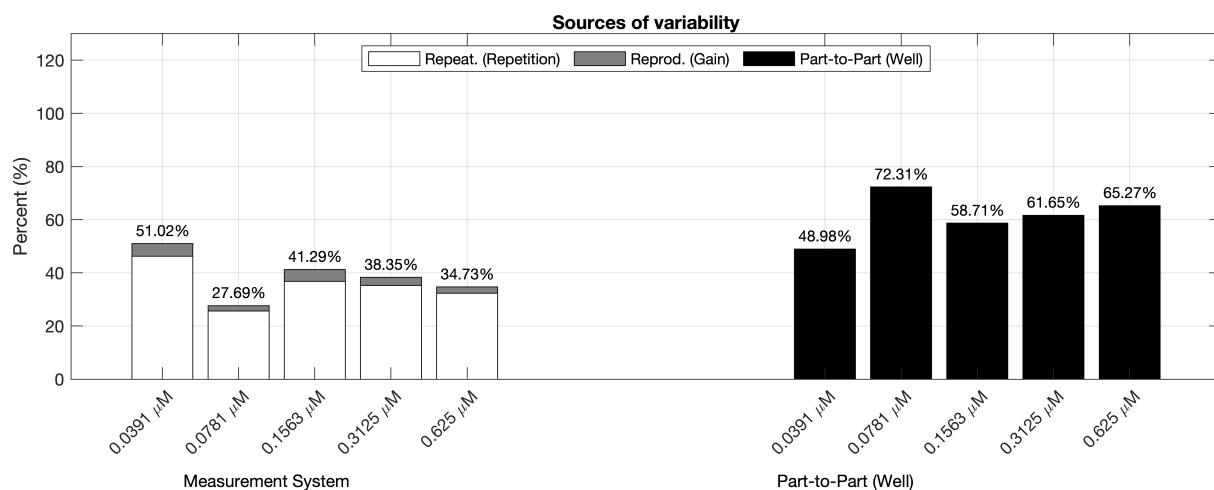
Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Reprod (Gain), $C = (0.3125)$	0.00255	3	0.00085	4.44	0.005
Replicates, $C = (0.3125)$	0.04356	4	0.01089	56.88	0
Error	0.0291	152	0.00019		
Total	0.07522	159			

Constrained (Type III) sums of squares.

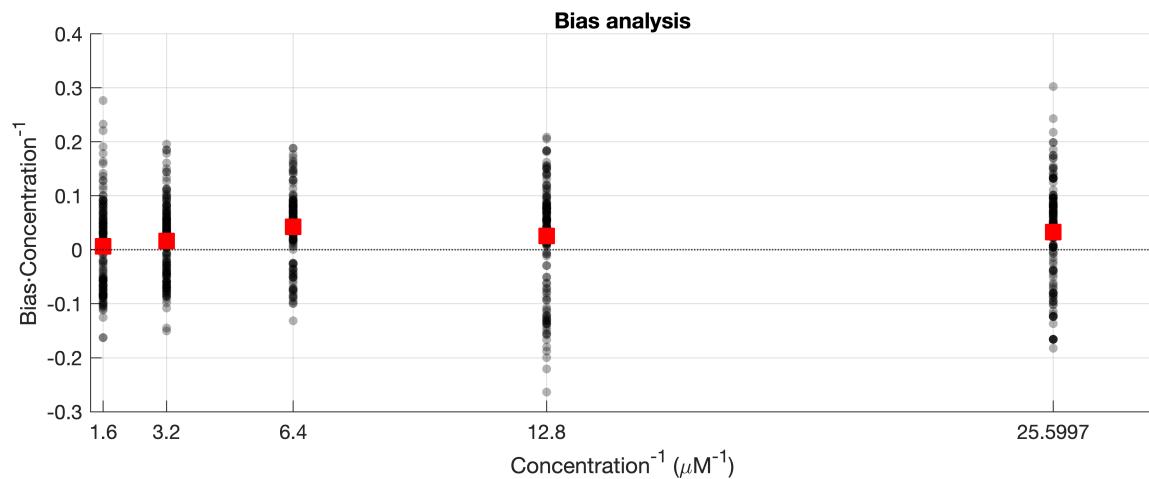
R & R Analysis on measurements for $C = 0.625$

Analysis of Variance					
Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Reprod (Gain), C = (0.625)	0.01077	3	0.00359	3.94	0.0097
Replicates, C = (0.625)	0.23901	4	0.05975	65.57	0
Error	0.13852	152	0.00091		
Total	0.3883	159			

Constrained (Type III) sums of squares.



B&L Analysis:



Linear regression model:

$$y \sim 1 + x_1$$

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	0.01764	0.0043869	4.0211	6.3413e-05
x1	0.00071458	0.00033201	2.1523	0.031675

Number of observations: 800, Error degrees of freedom: 798

Root Mean Squared Error: 0.082

R-squared: 0.00577, Adjusted R-Squared: 0.00453

F-statistic vs. constant model: 4.63, p-value = 0.0317

Contribution of model terms to the total bias variability:

Bias Model - linear term (%): 1.764 %

Bias Model - bias term (%): 4.9668 %

Confidence Intervals and Error metrics:

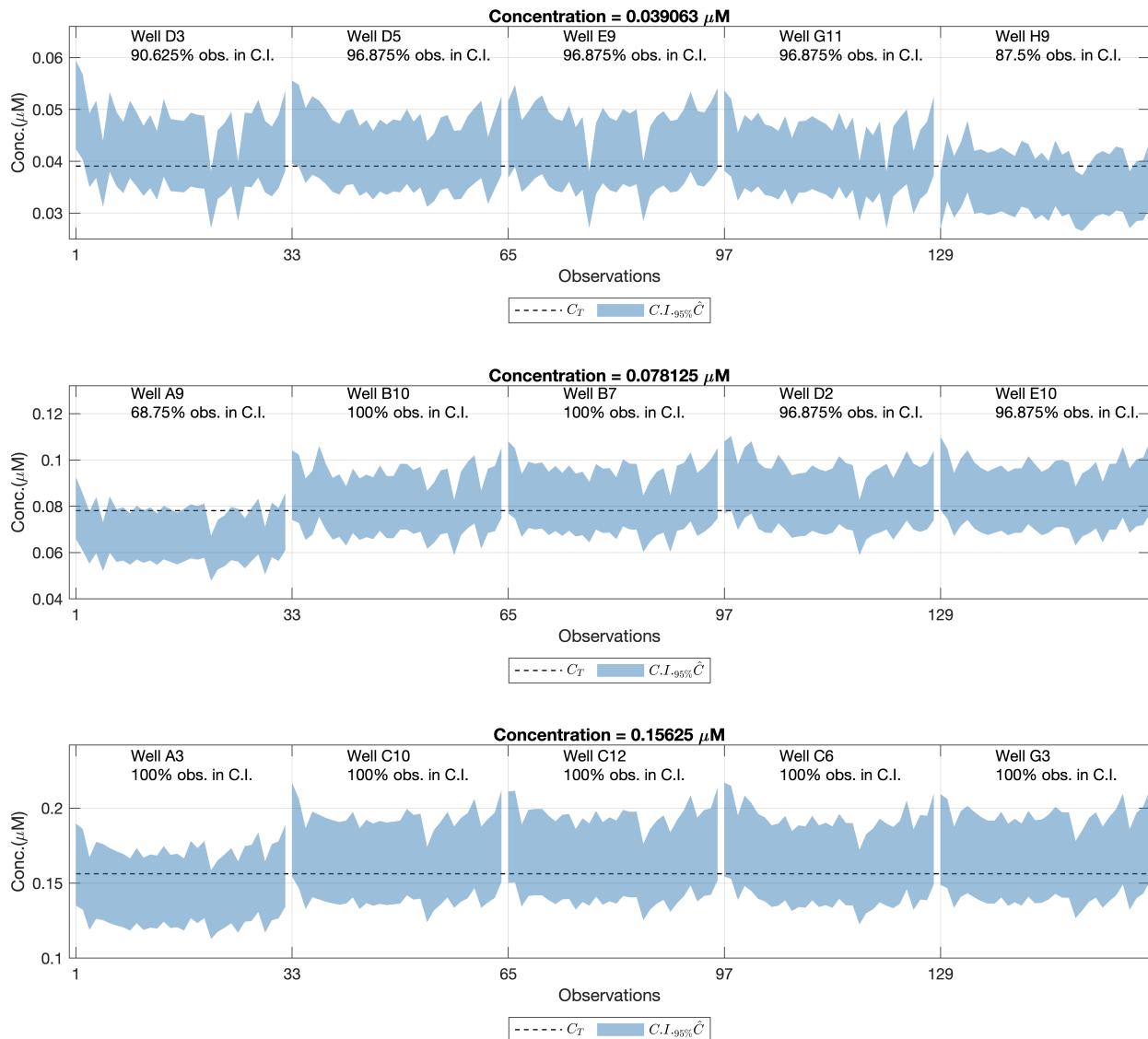
pctgeci: 96.3750

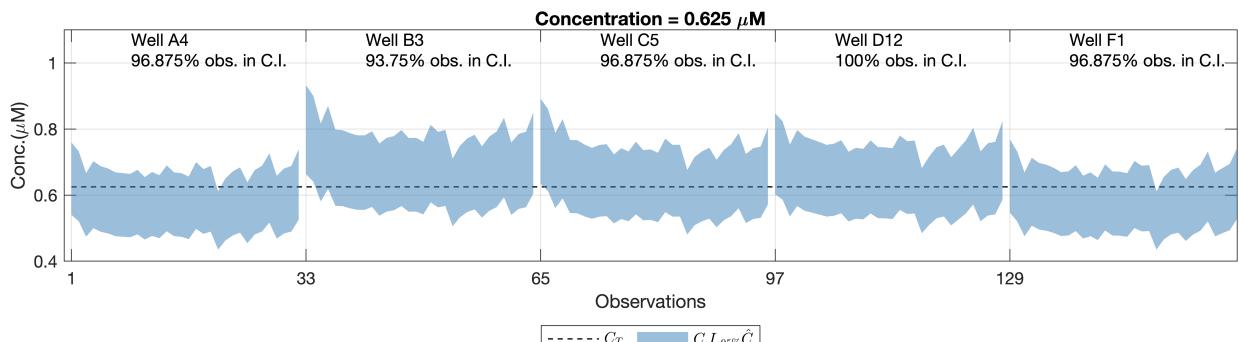
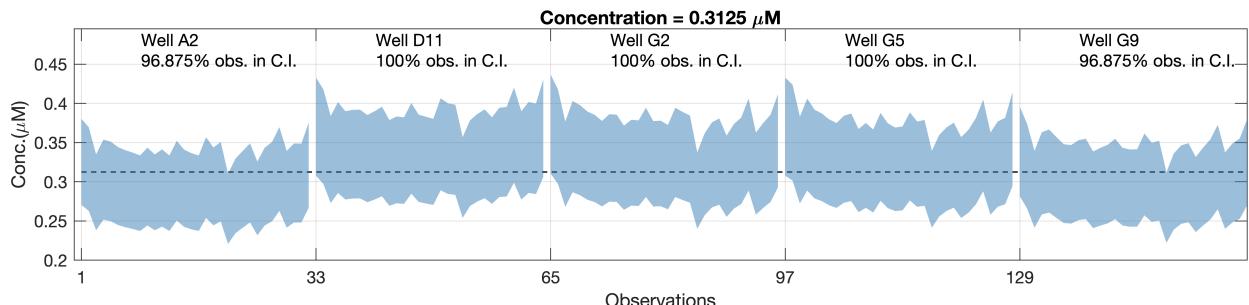
mse: 6.3208e-04

relerr: [800x1 double]

minrelerror: 6.1634e-04

maxrelerror: 0.3026





```
% Comparison between calibration-set and validation-set metrics
```

```
% load(strcat(dirdatasave,'cal_results.mat'))
perftable = table([calmetrics_PR2e2.mse;valmetrics_inrange.mse],...
[calmetrics_PR2e2.minrelerror;valmetrics_inrange.minrelerror]*100,... 
[calmetrics_PR2e2.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	6.0951e-04	0.0004	46.4025
2 Validation (within range)	6.3208e-04	0.0616	30.2628