Electronic Supplementary Material (ESI) for Analyst.

Characterization of the Impact of Mixing and Droplet Volumes on the Behavior of Microfluidic, Ion-Selective Droptodes

*Shannon P. Wetzler-Quevedo, Mark E. Meyerhoff, and Ryan C. Bailey**

Department of Chemistry, University of Michigan, Ann Arbor, Michigan, 48109, United States

Contents

1. **Data Analysis.** Data from the fluorescence of the oil phase, normalized to Normalized Fluorescence, was plotted against the logarithm of the potassium concentration in the sample. The data was then fit to a standard logistic fit (Equation S1):

$$Normalized\ Fluorescence = \frac{B-A}{1+(\frac{\log(K^+)}{C})^D} + A \qquad [S1]$$

Where A is the lower asymptote, B is the upper asymptote, C is the midpoint of the linear range and D is the slope of the linear range.

The code used to perform the analysis are included in Section 7 of the SI. The ImageJ code was run on a combined image with all of the individual points of the calibration included. Points were taken both in channel and in the background for control of lighting differences.

The CSV files outputted by ImageJ were run through the MATLAB code, in Section 7 of the SI, twice. The first time was using the outputted CSVs. This resulted in summary tables for both the in-channel and background points, which were then subtracted from one another. The code was then run again. Limits of Detection (LoD) and sensitivities were then determined from this data.

2. **Droplet Flow Rate Fitting.**
The relationship between total flow rate on the device (that is the sum of both the aqueous and oil flow rates), and the time spent on the device was calculated by dividing the volume traveled on the device by the total flow rate, or (Equation S2)

$$t = \frac{V_{channel}}{v} = \frac{0.096\ \mu L}{v} \qquad [S2]$$

Table S1. Relationship between Flow Rates and Time on Device

Total Flow Rate on Device (µL/min)	Time on Device (s)
2	6
4	1.4
8	0.7
16	0.4
32	0.2

Table S2. Fits of Droplet Flow Rate Data to Equation S1

Time on Device (s)	A	B	C	D	Log(LoD)
6	0.01 $\pm$ 0.5	1.2 $\pm$ 0.7	-4 $\pm$ 1	-9 $\pm$ 24	-4.470
1.4	0.01 $\pm$ 0.08	1.03 $\pm$ 0.08	-3.7 $\pm$ 0.2	-9 $\pm$ 4	-4.300
0.7	0.03 $\pm$ 0.2	1.0 $\pm$ 0.2	-3.5 $\pm$ 0.6	-7.4 $\pm$ 7	-4.403
0.4	0.01 $\pm$ 0.05	1.01 $\pm$ 0.05	-3.3 $\pm$ 0.1	-7 $\pm$ 2	-4.330
0.2	0.04 $\pm$ 0.14	1.0 $\pm$ 0.1	-3.3 $\pm$ 0.4	-10 $\pm$ 8	-3.683

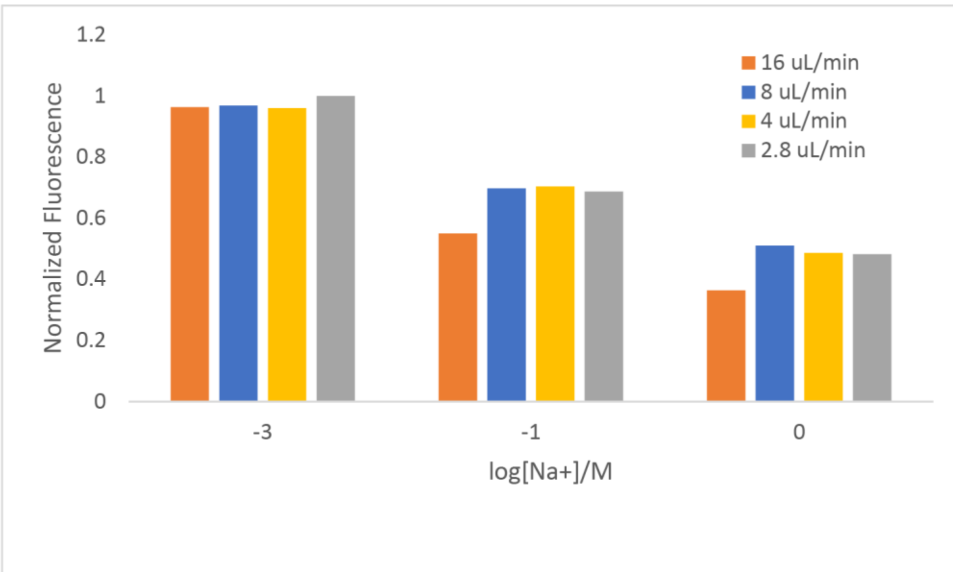

Figure S1. Na^+ Selectivity in a K^+ Selective Oil as Change Flow Rate. In a 1:1 flow ratio, at various total flow rates, aqueous droplets were flown in an oil carrier phase. The ideal value for a selective oil would be for 1-α to be 1 for all concentrations of Na^+. We see the first divergence from this when looking at 10^{-1} M Na^+. Note that the deviation from 1 is similar for the three lower flow rates, but greater for the highest one. This trend continues when looking at 1M Na^+.

3. Effects of Measurement Location Post-Droplet Formation on Calibration Curves.

Table S3. Fits of Droplet Location on Device to Equation S1

Location on Device (cm)	A	B	C	D	Log(LoD)
1	0.02 + 0.2	1.0 ± 0.2	-3.5 ± 0.8	-3 ± -2	-6.181
1.5	0.03 ± 0.08	0.99 ± 0.12	-3.4± 0.5	-3.4 ± 1.5	-6.774
3	0.04 ± 0.2	1.0 ± 0.1	-2.5±0.5	-4.4 ± 3.3	-4.36

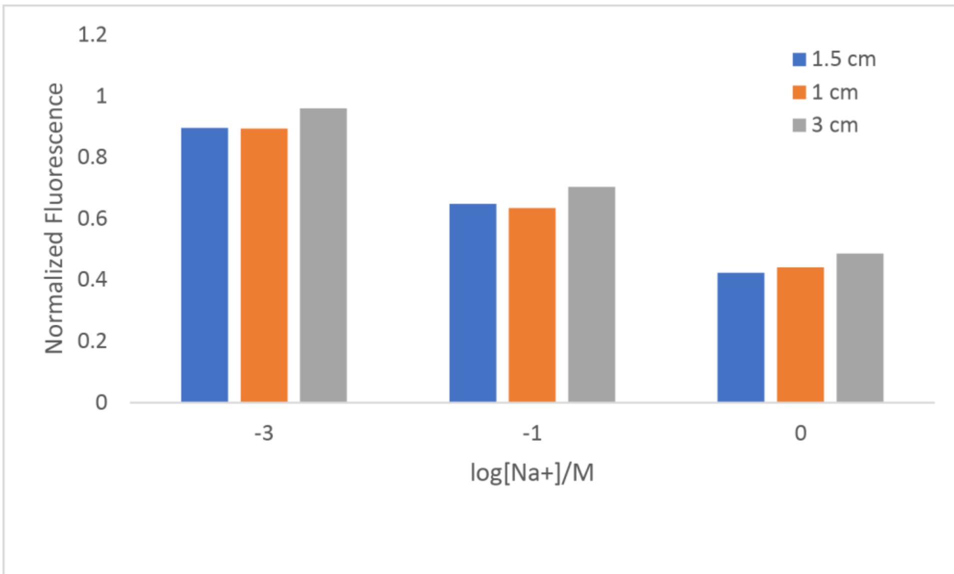

Figure S2. Na^+ Selectivity in a K^+ Selective Oil as Change Position. In a 1:1 flow ratio, at a 4 μL/min total flow rate, aqueous droplets were flown in an oil carrier phase. The ideal value for a selective oil would be for 1-α to be 1 for all concentrations of Na^+. We see the first divergence from this when looking at 10^{-3} M Na^+ when measuring at 1 and 1.5 cm. This is surprising as when looking at flow rate on device we do not see divergence until higher Na^+ concentrations. Note that the deviation continues when looking at higher Na^+ concentrations, where the 3 cm mark consistently has the closest to the ideal value of 1.

4. Effects of Mixing on Calibration Curves.

Table S4. Fits of Serpentine Channel on Device to Equation S1

Location on Device (cm)	A	B	C	D	Log(LoD)
1.5	0.09 ± 0.08	0.90±0.09	-2.7 ± 0.3	-4 ± 2	-4.3685
3	0.2 + 0.2	1.0 ± 0.3	-3 ± 1	-3 ± 2	-5.019

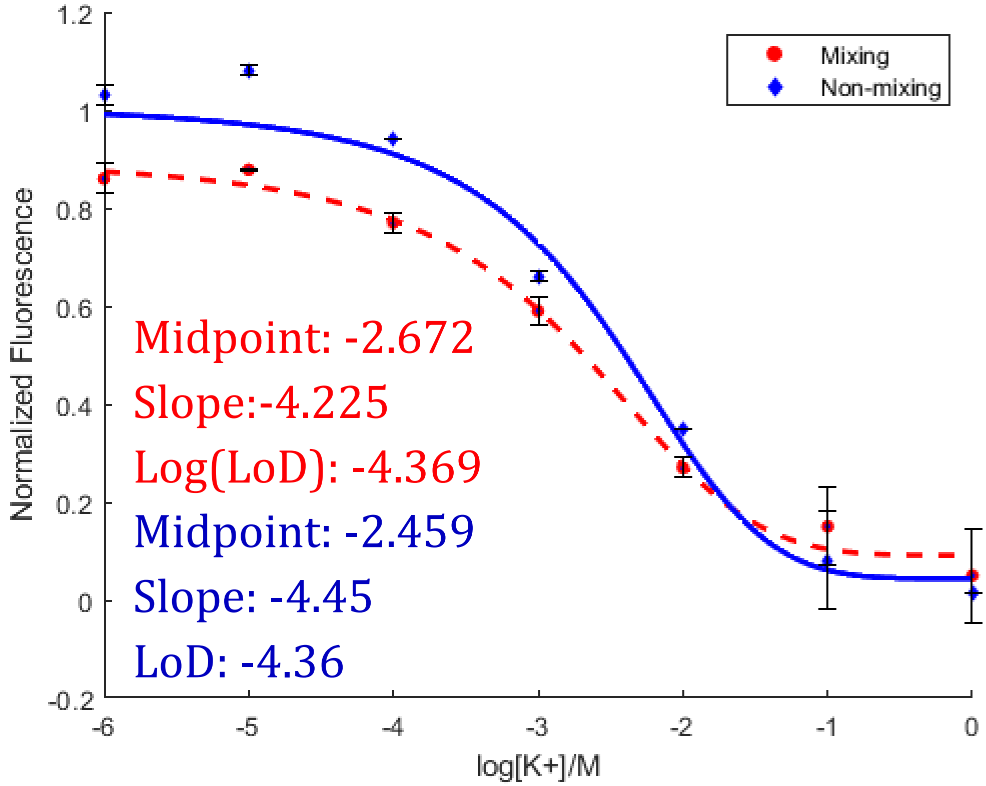

Figure S3. Comparing the 1.5 cm measurement on mixing device to the 3 cm measurement on linear device. The slopes and midpoints are similar to one another, but more interestingly the limits of detection are within 0.01 log units of one another. This potentially suggests that the 1.5 cm mixing device is able to equilibrate to the same point as the 3 cm linear device, in half the time.

5. Effect of Droplet Size on Calibration Curves.

Table S5. Fitting changing droplet and spacing size data to Equation S1

$V_{oil}/(V_{oil}+V_{aqueous})$	A	B	C	D	Log(LoD)
0.25	0.1 ± 0.1	1.1 ± 0.1	-3.2 ± 0.3	-8 ± 6	-4.502
0.5	0.04 ± 0.2	1.0 ± 0.1	-2.5 ± 0.5	-4 ± 3	-4.887
0.75	0.1 ± 0.2	1.1 ± 0.2	-3.1 ± 0.6	-4 ± 3	-5.450

6. Code Used in Image J and MATLAB.

ImageJ: These codes were used to combine images into one file and then analyze their fluorescence.

```
macro "Combine [0]"{
path = getDirectory("Choose a data folder");
setBatchMode(true);
list = getFileList(path);
for (i = 0; i < list.length; i++)
	open(list[i]);
x = getNumber("Number of samples being measured please",8);
y = getNumber("Number of measurement points per sample please",2);
for(i=1;i<=y;i++)
{
		title = getString("What do you want this to be called", "Booger");
		for(j=1; j<x; j++)
		{
				run("Combine...");
		}
		saveAs("tiff",path+title);
}
title2 =  getString("What do you want this to be called", "Booger");
for(k=1; k<y; k++)
{
run("Combine...");
}
saveAs("tiff",path+title2);
}

macro "Time Point [b]"{
title = getTitle();
path = getDirectory("Where do you want to save enhanced image")
path2 = getDirectory("Choose where to put logs")
selectWindow("" + title +"");
run("Enhance Contrast...", "saturated=0.3 process_all");
saveAs("gif",path+title);
z = getNumber("Number of samples please",6);
A = getNumber("What rectangle width do you want",100);
B = getNumber("What rectangle height do you want", 25);
for(j=1; j<=z; j++)
{
name = getString("How do you want to name the sample", "T");
setSlice(1);
x = getNumber("What is the x value?",100);
y = getNumber("What is the y value?",25);
makeRectangle(x, y, A, B);
for(i=1; i<=nSlices(); i++)
{
```

```
run("Measure");
run("Next Slice [>]");}
   headings = split(String.getResultsHeadings);
   i=1;
   for (row=0; row<nResults; row++) {
      line = "";
      for (col=1; col<lengthOf(headings); col++)
      line = line + getResult(headings[col],row) + "  ";
      print(line);
   }
 saveAs("Text", path2+name+"Log.csv");
 IJ.deleteRows(0,nResults);
 print("\\Clear");
}
close();
}
```

MATLAB: This code was used to process the data output by the ImageJ code.

```
close all
clear all
dir1='C:\Users\swetzler\Desktop\01312021\CO3\AVI\Logs'
cd 'C:\Users\swetzler\Desktop\01312021\CO3\AVI\Logs'
files = dir('*.csv');
N = length(files);    % total number of files
mkdir New;
SumEnd=[    ];
SumT=[    ];
SumMid=[    ];
SumPB=[     ];
SumFin=[     ];
SumEnd1=[    ];
SumT1=[    ];
SumMid1=[    ];
% loop for each files
for i = 1:N
    thisfile = files(i).name
    if strlength(thisfile)>4
        str=strcat(dir1,'\',thisfile);
        filename = erase(string(thisfile),".csv");
        %open file
        fid = fopen(str,'rt');
        %figure out how many columns are there
        firstline = fgetl(fid);
        ncol = 4;
    %reset to beginning of file
        fseek(fid,0,0);
    %read data
        indata = textscan(fid,repmat('%f%*n',1,ncol),'Delimiter','
','CollectOutput',1);
        V = [size(indata{1}), 1]; V(find(V == 1, 1)) = numel(indata);
        A = reshape(horzcat(indata{:}), V);
        nrow=size(A,1);
        for(i=1:nrow)
            A(i,4)=i;
        end
```

```
        n=1;
        k=[];
        for(i=1:nrow)
            if(A(i,1)==0)||i>200
                k(n)=i;
                n=n+1;
            end
        end
        for(j=size(k,2):-1:1)
           A(k(j),:)=[];
        end
        CondA=A(:,[1 4]);     %close file
        fclose(fid);
        D=CondA;
        D(:,2)=CondA(:,1);
        D(:,1) = CondA(:,2);

        writematrix(D,dir1+"\New\new"+thisfile);

        B=D;
        r=size(B);
        c=r(2);
        r=r(1);
        skipped=0;


    %identify locations of largest peak in each spectrum
        [Z,z] = max(D,[],1);
        highest=(B(z(:),1));
        ordered=sort(highest);
        Tallest=unique(ordered);
        T=size(Tallest,1);
        counter=zeros(T);
        check=0;
        plus=0;
        minus=0;
        LDE=0;
        RDE=0;
        Ld=0;
        Rd=0;
        Peaks=0;
        Location=0;
        plot(B(1:r-1, 1), CondA(2:r,1),'DisplayName', 'average');

        [pks, locs] =
findpeaks(B(:,2),'MinPeakDistance',1,'MinPeakHeight',0.85*max(B(:,2)));
        text(locs+.02,pks,num2str(locs));
        filename = erase(filename,"Log");
        filename1 = string(filename);
        xlabel('Frame')
        ylabel('Fluorescence')
        saveas(gcf,dir1+"\New\cal"+filename1+".png");
    %drops
        B2=-D;
        r1=size(B2);
        c=r1(2);
```

```
        r1=r1(1);
        skipped=0;


    %identify locations of largest peak in each spectrum
        [Z,z] = max(-D,[],1);
        highest1=(B2(z(:),1));
        ordered1=sort(highest1);
        Tallest1=unique(ordered1);
        T1=size(Tallest1,1);
        counter1=zeros(T1);
        check1=0;
        plus1=0;
        minus1=0;
        LDE1=0;
        RDE1=0;
        Ld1=0;
        Rd1=0;
        Peaks1=0;
        Location1=0;
        C2=-CondA(2:r1,1);
        plot(B2(1:r1-1, 1), C2,'DisplayName', 'average');
        [pks2, locs2] =
findpeaks(B2(:,2),'MinPeakDistance',1,'MinPeakHeight',1.1*max(B2(:,2)));
        text(locs2+.02,pks2,num2str(locs2));
        diff = mean(pks)-abs(mean(pks2));
        diffstdv=sqrt(std(pks)^2+std(pks2)^2);
        filename = erase(filename,"Log");
        filename1 = string(filename);
        saveas(gcf,dir1+"\New\cal"+filename1+"drops.png");
        if contains(filename1,"T")
            filename = erase(filename,"T");
            measurement=[filename mean(pks) std(pks) abs(mean(pks2))
std(pks2) diff diffstdv];
            SumT=[SumT;measurement];



        elseif contains(filename1,"Mid")
            filename = erase(filename,"Mid");
            measurement=[filename mean(pks) std(pks) abs(mean(pks2))
std(pks2) diff diffstdv];
            SumMid=[SumMid;measurement];
        elseif contains(filename1,"PB")
            filename = erase(filename,"PB");
            measurement=[filename mean(pks) std(pks) abs(mean(pks2))
std(pks2) diff diffstdv];
            SumPB=[SumPB;measurement];
        elseif contains(filename1,"End")
            filename = erase(filename,"End");
            measurement=[filename mean(pks) std(pks) abs(mean(pks2))
std(pks2) diff diffstdv];
            SumEnd=[SumEnd;measurement];
        %measurement1=[filename mean(minima) std(minima)];
        %SumEnd1=[SumEnd1;measurement1];
        else
            filename = erase(filename,"Fin");
```

```
            measurement=[filename mean(pks) std(pks) abs(mean(pks2))
std(pks2) diff diffstdv];
            SumFin=[SumFin;measurement];
        end
    end
end
writematrix(SumEnd,cd+"\New\Endcal.csv")
%writematrix(SumT,cd+"\New\Tcal.csv")
writematrix(SumMid,cd+"\New\Midcal.csv")
% writematrix(SumPB,cd+"\New_05022020\PBcal.csv")
writematrix(SumFin,cd+"\New\Fincal.csv")
%writematrix(SumEnd1,cd+"\New_Diff\Endcal_drops.csv")
%writematrix(SumT1,cd+"\New_Diff\Tcal_drops.csv")
%writematrix(SumMid1,cd+"\New_Diff\Midcal_drops.csv")
x=str2double(SumEnd(:,1));
y=str2double(SumEnd(:,6));
stddev=str2double(SumEnd(:,7));
errorbar(x,y,stddev, 'bx')
x=str2double(SumEnd(:,1));
y=str2double(SumEnd(:,2));
stddev=str2double(SumEnd(:,3));
errorbar(x,y,stddev, 'bo','MarkerFaceColor','b')
xlabel('log(CO32-)/[M]')
ylabel('Fluorescnce')
saveas(gcf,dir1+"\New\cal_fig.png")
% xT=str2double(SumT(:,1));
% yT=str2double(SumT(:,6));
% stddevT=str2double(SumT(:,7));
% hold on
% errorbar(xT,yT,stddevT,'rx')
% saveas(gcf,dir1+"\New\cal.png")
hold on
xM=str2double(SumMid(:,1));
yM=str2double(SumMid(:,2));
stddevM=str2double(SumMid(:,3));
errorbar(xM,yM,stddevM,'mx')
saveas(gcf,dir1+"\New\cal_fig.png")
hold on
xF=str2double(SumFin(:,1));
yF=str2double(SumFin(:,2));
stddevF=str2double(SumFin(:,3));
errorbar(xF,yF,stddevF,'gx')
saveas(gcf,dir1+"\New\cal_fig.png")
```