# An integrated microfluidic device for two-dimensional combinatorial dilution

## Supplementary Information

*Yun-Ho Jang[a,b], Matthew J. Hancock[a,b], Sang Bok Kim[a,b], Šeila Selimović[a,b], Woo Young Sim[a,b], Hojae Bae[a,b], and Ali Khademhosseini[a,b,c]**

[a] Center for Biomedical Engineering, Department of Medicine, Brigham & Women's Hospital, Harvard Medical School , Boston, MA 02115, USA

[b] Harvard-MIT Division of Health Sciences and Technology, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

[c] Wyss Institute for Biologically Inspired Engineering, Harvard University, Boston, MA 02115, USA

* E-mail: A. Khademhosseini: alik@rics.bwh.harvard.edu

## Supplementary Methods

### Methods I. Estimate of inlet velocity for CFD simulation from experimental conditions.

The inlet speed 0.333 m $s^{-1}$ used in the CFD simulations was estimated from the experimental conditions for the 10 × 10 device at 80 kPa as follows. The volume injected into the wells in one column in one cycle at 80 kPa was 1.51 times the well volume 21.2 nL (see results section in main text on MAI characterization). Each channel between wells was ~700 um long with a 1200 $\mu m^2$ cross-sectional area. Thus, the ten channels in a column of wells had a total volume of approximately 8.4 nL. The total volume injected into a column of wells was therefore between 1.5 × 21.2 = 31.9 nL and 31.9 + 8.4 = 40.3 nL. In our simulations, we injected ~40 nL over ~0.1 s, which corresponds to a flow rate of ~36 μL $min^{-1}$ and a flow speed of ~0.333 m $s^{-1}$ (based on the 1200 $\mu m^2$ semi-circular channel cross-section). The ~40 nL is probably an overestimate of the total injected volume, but was sufficient to obtain CFD results that closely mimicked the experiments. Note that to derive our analytic formula, we neglected the effects of the channels and used only the volume injected into the wells, 31.9 nL.

### Methods II. Normalization of fluorescent intensities

In the experiments related to Fig. 4 (and Supp Fig. S10), we normalized all green fluorescent intensities with respect to the left-most reservoir, which had the same depth as the deep wells and contained 100% green solution and undetectable red solution. Similarly, all red fluorescent intensities were normalized by the intensity of the right-most reservoir. The blue intensities were normalized with respect to the maximum intensity measured in each pre-filled column, prior to active injection. In this way, we attempted to scale all intensities by values related to the pure input solutions.

# Supplementary Tables

**Table ST1.** Photolithography protocol for preparation of control and flow layers.

| | Control layer | Flow layer | |
|---|---|---|---|
| Substrate | 4" silicon wafer | 4" silicon wafer | |
| | Control channels | Flow channels | Deep wells |
| PR type | SU-8 2050 | AZ4620 | SU-8 2150 |
| Thickness (μm) | 75 | 20 | 300 |
| Spin coat | 5 sec @ 500 rpm<br>30 sec @ 2000 rpm | 10 sec @ 500 rpm<br>30 sec @ 1000 rpm | 10 sec @ 500 rpm<br>30 sec @ 1300 rpm |
| Soft bake | 20 min @ 95 °C | 20 min @ 95 °C | 10 min @ 65 °C<br>75 min @ 95 °C |
| UV exposure ($mJ\ cm^{-2}$) | 160 | 360 | 480 |
| Post exposure bake | 5 min @ 95 °C | - | 5 min @ 65 °C<br>30 min @ 95 °C |
| Development | 15 min @ PM acetate | 3 min @ 4:1 diluted AZ 400 MIF | 30 min @ PM acetate |
| Hard bake | - | 3 min @ 110 °C<br>5 hr @ 150 °C | - |

# Supplementary Videos

**Video S1. One-dimensional (1D) gradient generation by microfluidic active injection (MAI).** Step-by-step MAI procedure: solution loading; reservoir isolation; opening of channel valves; pressurization of the reservoir membranes which generates a 1D gradient along each column of wells; deep well isolation; and visualization of the gradient across the deep well array. The video was captured on a Nikon TE2000-U microscope with VirtualDub software at 15 frames per second.

# Supplementary Figures

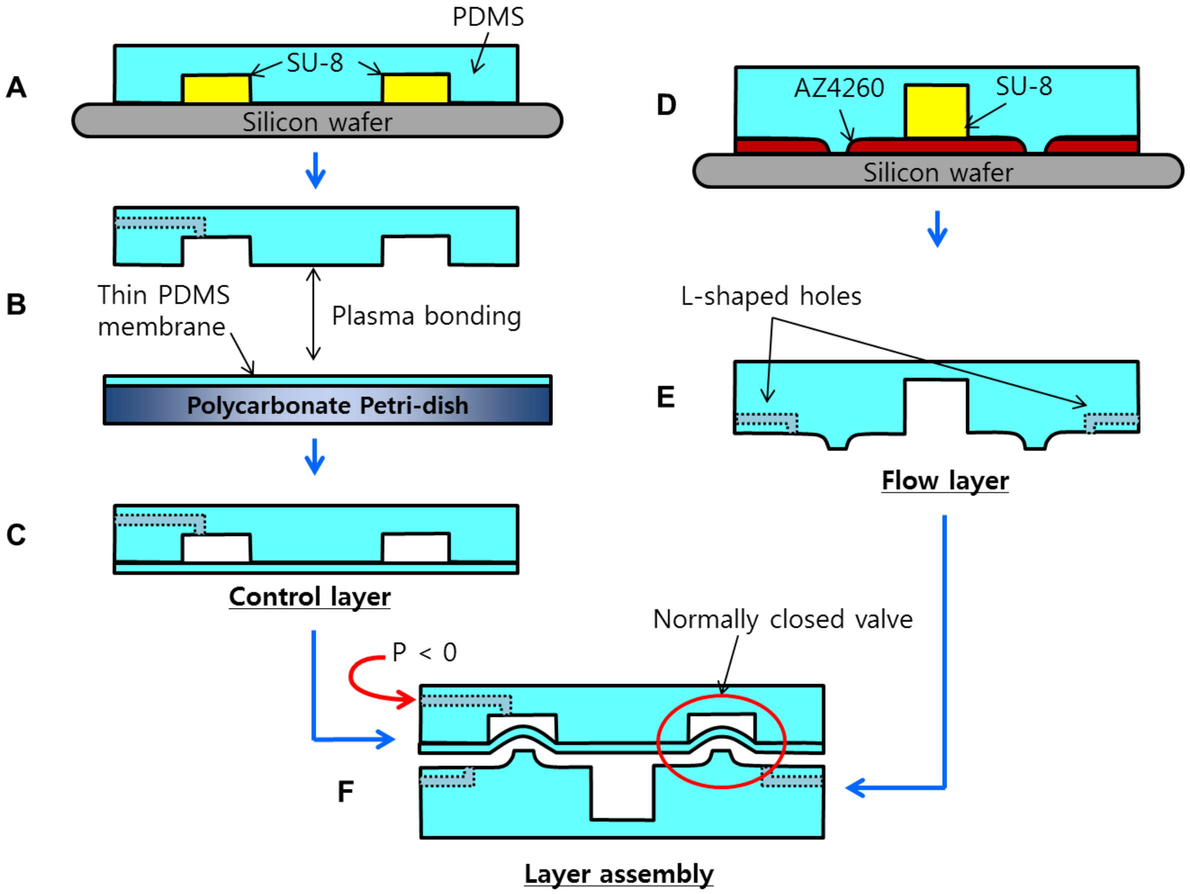



**Figure S1. Fabrication process.** (A) Control channel fabrication using SU-8 2050. (B) After moulding, the L-shaped connection holes were formed for connection to a pressure source. A thin PDMS membrane (15 μm) was prepared by spin-coating a polycarbonate Petri dish. The PDMS membrane was bonded to the control channel after treatment with oxygen plasma. (C) The membrane and control layer were peeled off from the Petri dish after baking for 60 min at 80 ℃. (D) The flow layer was fabricated using positive photoresist (AZ4620) for flow channels, and negative photoresist (SU-8 2150) for deep wells. (E) L-shaped connection holes were also formed in the fluidic channel layer after moulding. (F) The flow and control layers were bonded after oxygen plasma treatment at 80 °C for 30 min. During this process we applied negative pressure to the control channels in order to prevent permanent bonding in the area of normally-closed valves.

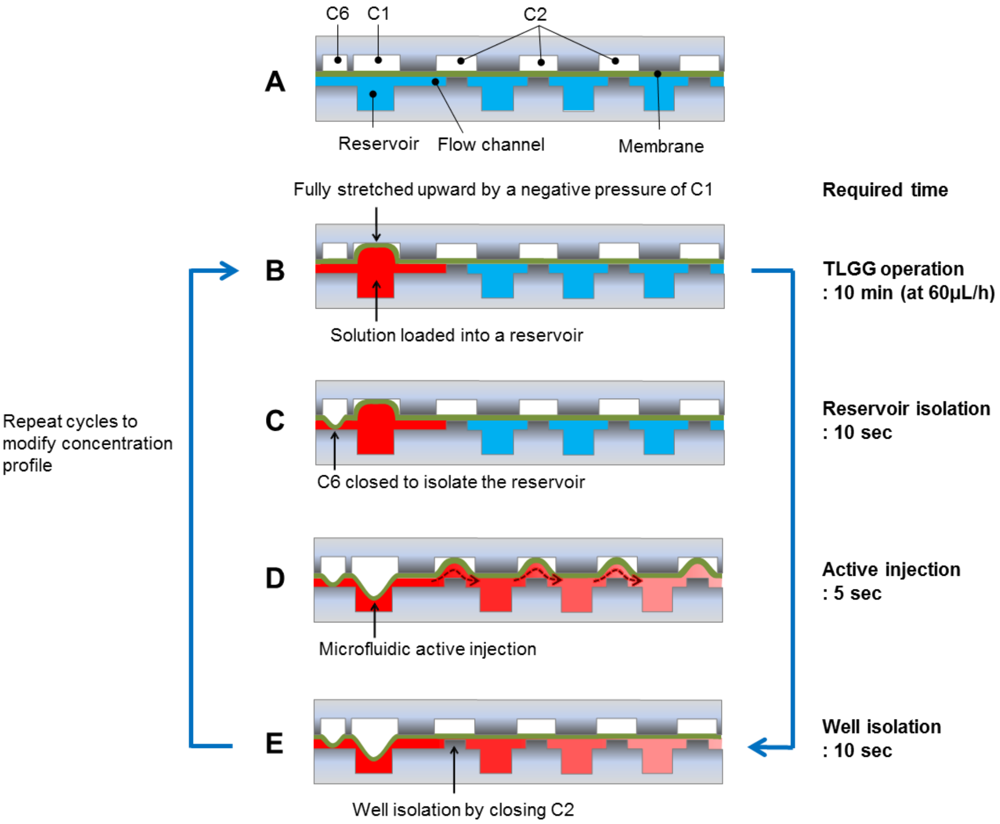



**Figure S2. Schematic of operation protocol for microfluidic active injection (MAI).** Shown in each diagram is a cross-section of our device with a reservoir, wells, valve membranes, and flow and control channels. (A) All wells and flow channels were prefilled with solution C. (B) Solution A was loaded into a reservoir and negative pressure was applied to control channel C1 to stretch the membrane fully upward to force all reservoir volumes to be the same. This corrects for the possibly different natural curvatures of the membranes. (C) Control channel C6 was closed to isolate the solution in the reservoir. (D) After C2 was opened by applying negative pressure, positive pressure was applied to C1 to push the stored solution from the reservoir into the pre-filled well array. (E) Closing C2 isolated all deep wells to maintain their composition.

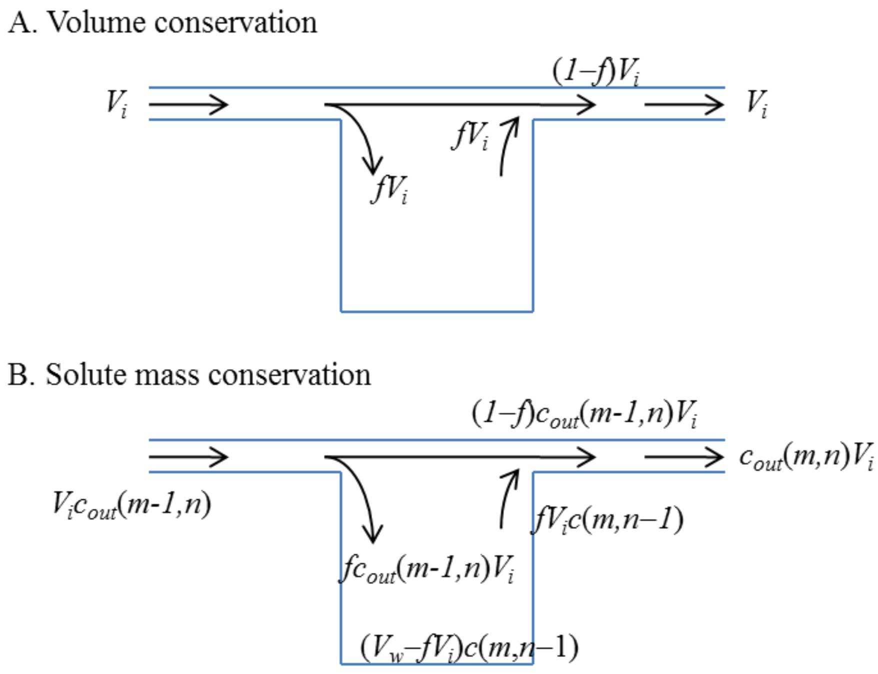



**Figure S3. Schematic for derivation of approximate analytic formula.** (A) Volume conservation. During cycle $n$, a volume $V_i$ flows into well $m$ and a fraction $f$ of that stays in the well, where $0 < f < 1$. Since the fluid is incompressible, a fraction $fV_i$ is also ejected from the well. (B) Solute mass conservation. A mass $fV_ic_{out}(m\text{-}1,n)$ enters well $m$ from well $m$-1, while a mass $fV_ic(m,n\text{-}1)$ is ejected. The well also has a solute mass $(V_w\text{-}fV_i)c(m,n\text{-}1)$ remaining from the previous injection cycle.

Electronic Supplementary Material (ESI) for Lab on a Chip

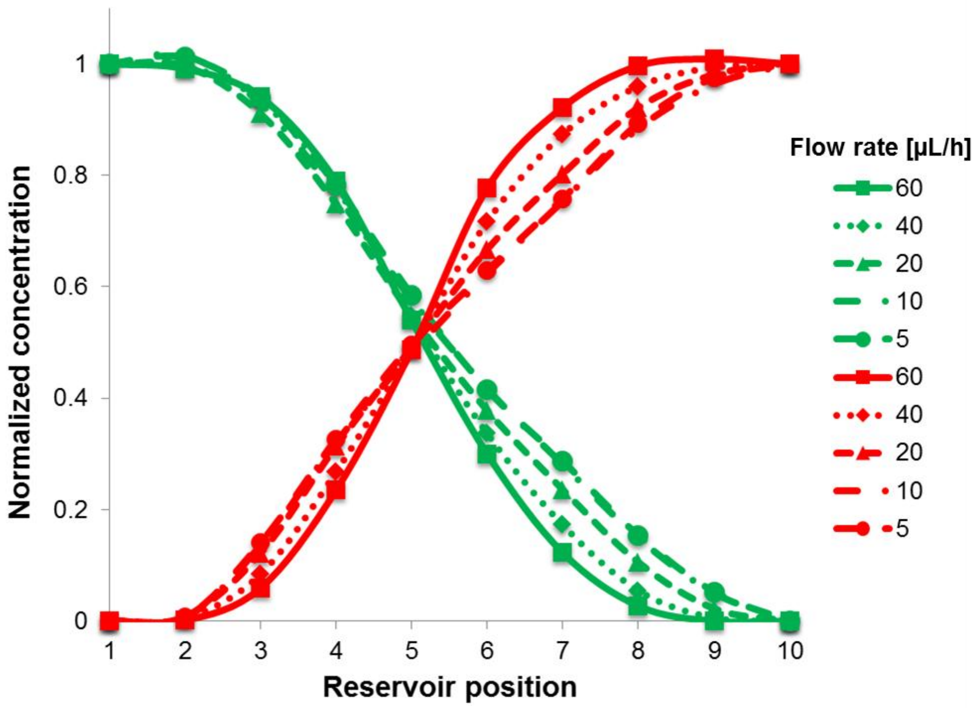




**Figure S4. Tree-like gradient generator (TLGG) characterization.** Discrete concentration gradients of solutions A and B were generated by the TLGG and stored in the ten reservoirs. Solution A was fluorescein sodium salt (green) and solution B was sulforhodamine 101 (red). Low TLGG input flow rates (<5 µL $h^{-1}$) generate near-linear gradients but require longer stabilization and reservoir filling times. At a TLGG input flow rate of 60 µL $h^{-1}$ per stream, the reservoir was filled within 30 seconds, but we waited 10 minutes before isolating the filled reservoirs to ensure that the solutions A and B were properly mixed and the solution concentrations in each reservoir had stabilized.

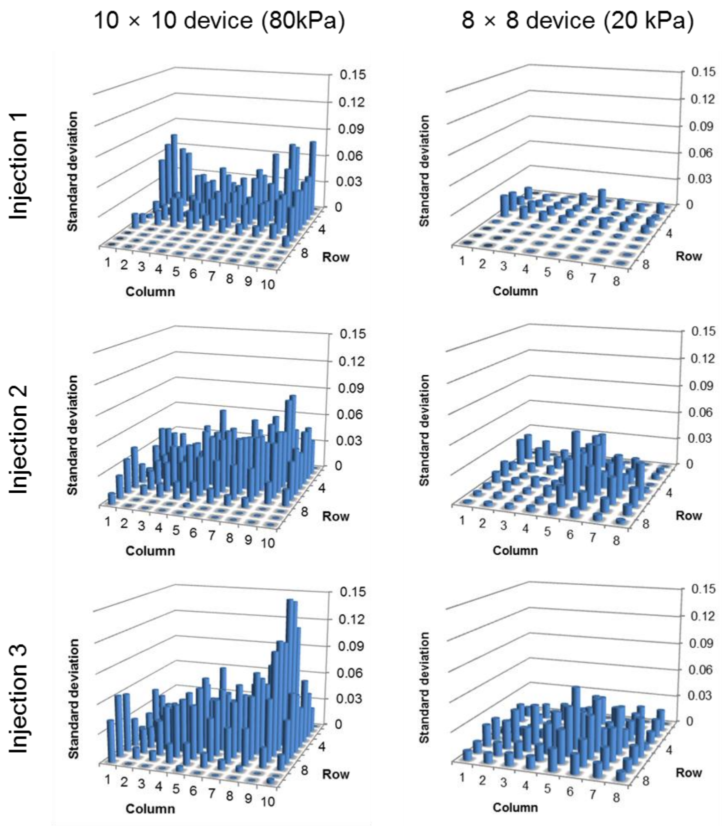



**Figure S5. Run-to-run variation of well concentrations.** The standard deviations (over three repetitions) of the average well concentrations plotted in Figure 2 of the main text, shown for the 10 × 10 device operated at 80 kPa and the 8 × 8 device operated at 20 kPa. The 8 × 8 device had significantly lower standard deviations than the 10 × 10 device. The design differences between the devices required different ranges of injection pressures to be used for their operation; the behaviour of the 10 × 10 device did not vary significantly with injection pressure (Fig. S7).

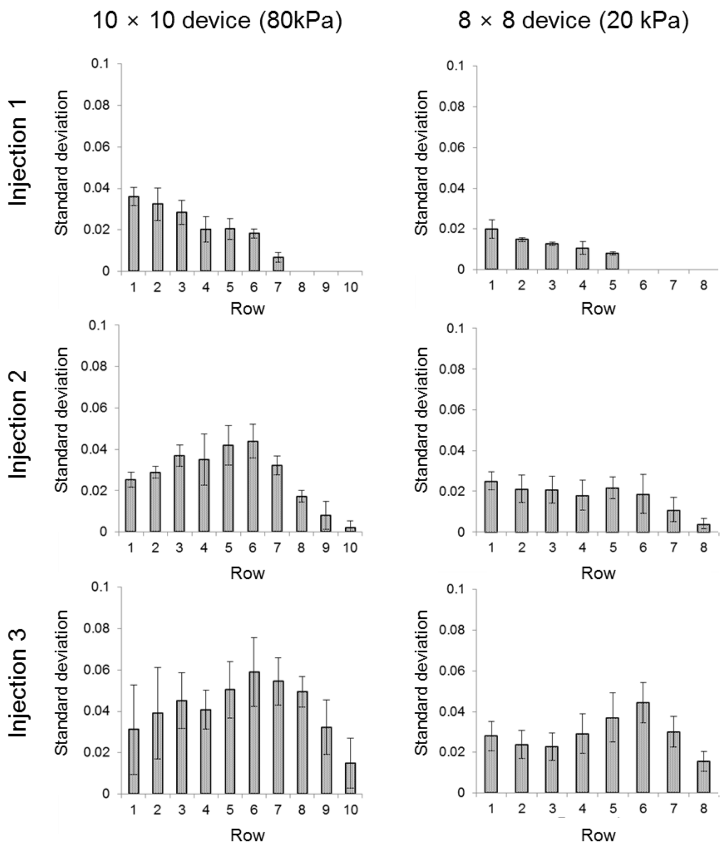



**Figure S6. Column-to-column variation of well concentrations.** The standard deviations of the column averages of well concentrations plotted in Figure 2 of the main text are shown for the 10 × 10 device operated at 80 kPa and the 8 × 8 device operated at 20 kPa. Error bars represent the standard deviation over three repetitions of the standard deviations of the column-averaged well-concentrations. The 8 × 8 device had significantly lower column-averaged standard deviations than the 10 × 10 device.

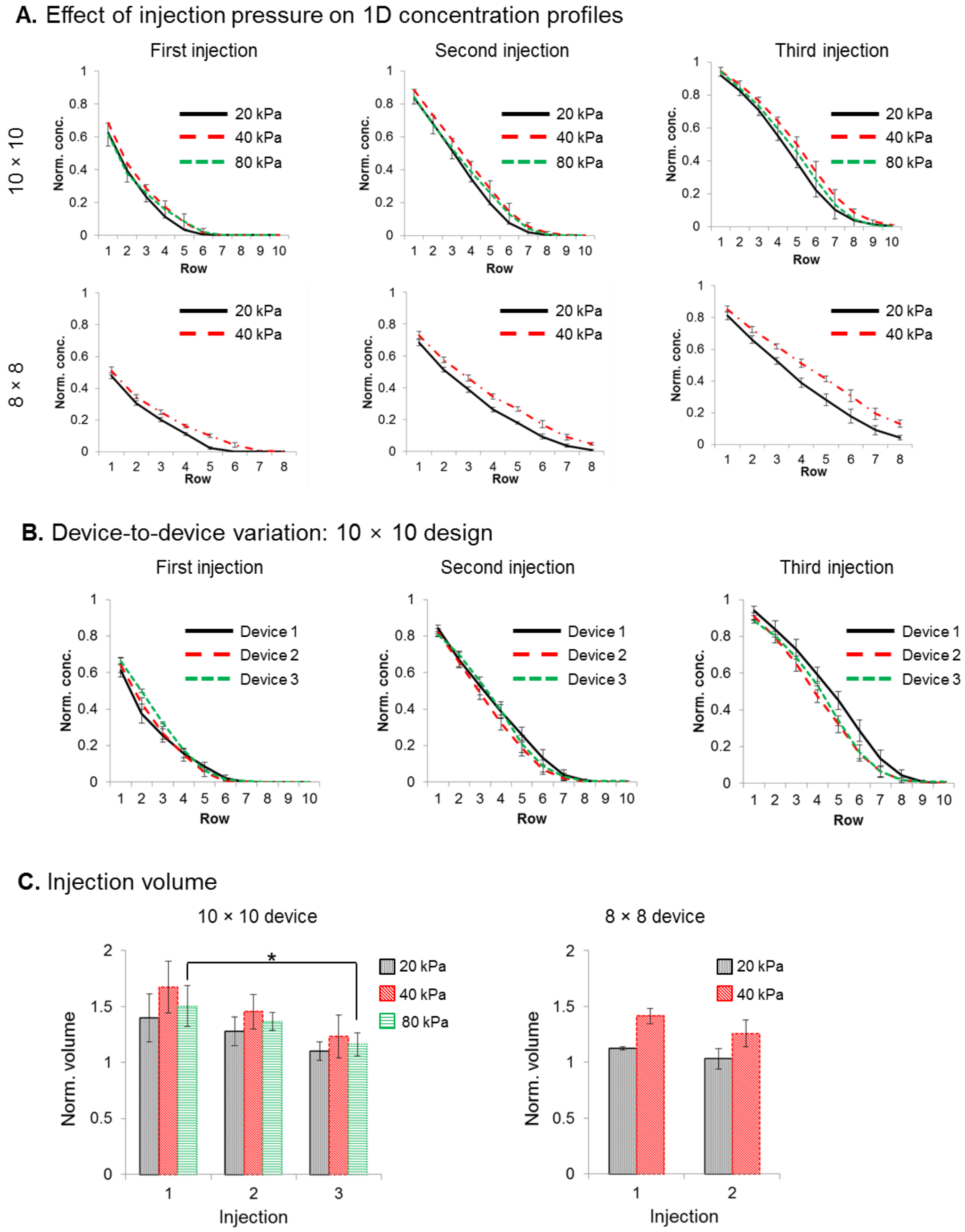



**Figure S7. Pressure and device design/fabrication effects on well concentrations and measured injection volume.** (A) Effect of injection pressure on column-averaged well concentrations averaged over three repeated experimental runs on the same device, for both the 8 × 8 and 10 × 10 designs. Error bars are shown for the 10 × 10 device operated at 80 kPa and for the 8 × 8 device at two pressures, and indicate the standard deviation over three repetitions. (B) The well concentrations in experiments repeated on different devices show nominal variation, more-so for later injections. Error bars indicate the standard deviation over three repetitions. For A and B, curves were formed from lines connecting the discrete normalized concentration at each well. (C) Injected volumes normalized by the well volume (21.2 nL for the 10 × 10 device and 31.8 nL for the 8 × 8 device) for different injection pressures. Volumes were calculated from the sum of the well (fluorescent) intensities across the array (see main text for details). The difference between the measured volumes were not significant, except where noted by (*), where $p < 0.05$. Volumes from the third injection in the 8 × 8 device were not calculated because by then the fluorescent solution had moved past the last row of wells.

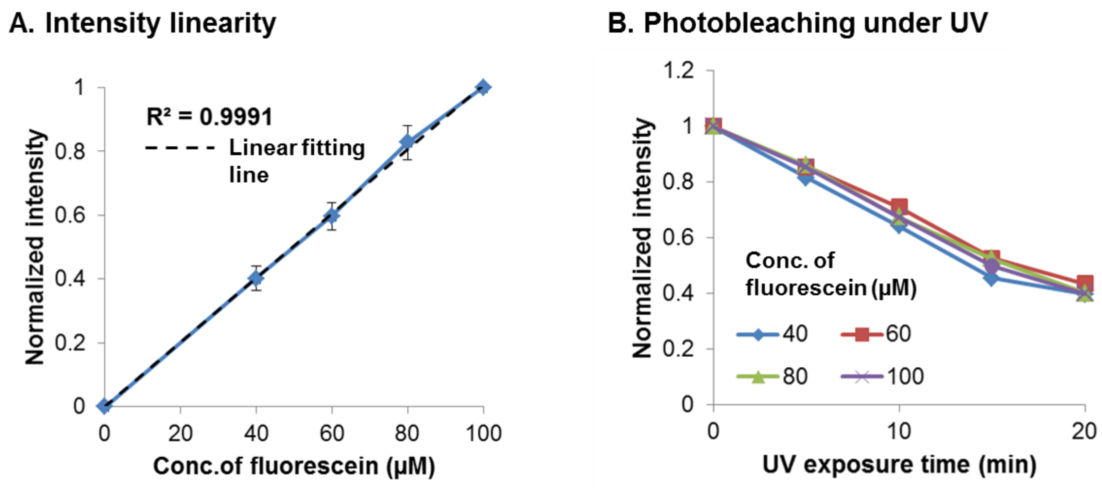



**Figure S8. Intensity-concentration linearity and photobleaching rate of fluorescein.** (A) Normalized intensity with respect to the concentration of fluorescein. The intensity is linearly proportional to concentration for concentrations less than 100 μM. Error bars represent the standard deviation over three experimental runs of one 10 × 10 device. (B) Photobleaching of fluorescein under continuous UV exposure. 10 x 10 well array devices were pre-filled with solutions containing different concentrations of fluorescein. The average intensities decreased to 85% and 68% after 5 min and 10 min, respectively.

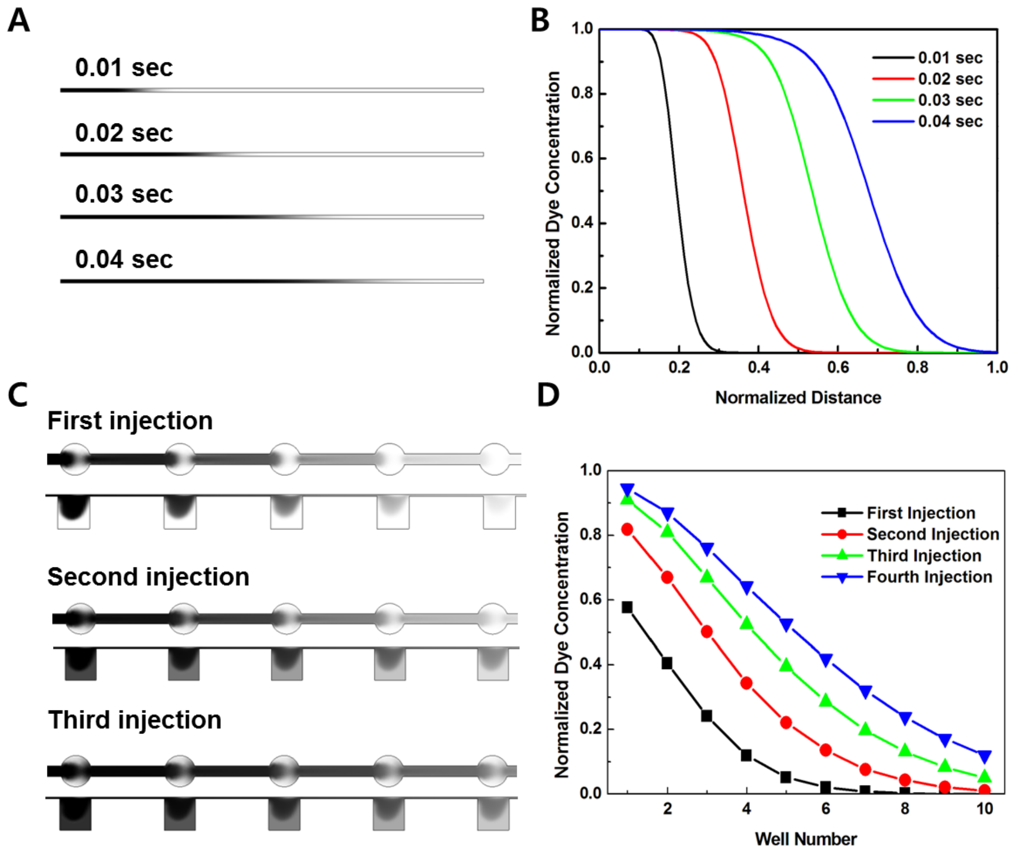



**Figure S9. Concentration profile in a simple channel vs. a channel and well geometry.** (A) Concentration profiles along an axial cross-section of a rectangular channel (20 µm high, 100 µm wide, 1 cm long). (B) Cross-sectionally averaged gradient profiles in the rectangular channel at different times. The gradient spanned less than half the channel length. (C) Concentration profiles along an axial cross-section of the channel-well geometry after successive injections. The channel was 20 µm high and 100 µm wide and the wells were 300 µm deep and 300 µm in diameter. The total channel length was 1 cm (same as the simple rectangular channel); only half that is shown in the image. (D) Average concentration in each well along the channel-well geometry. The profile in the channel-well geometry is longer than that in the rectangular channel due to sequential dilution in each well.

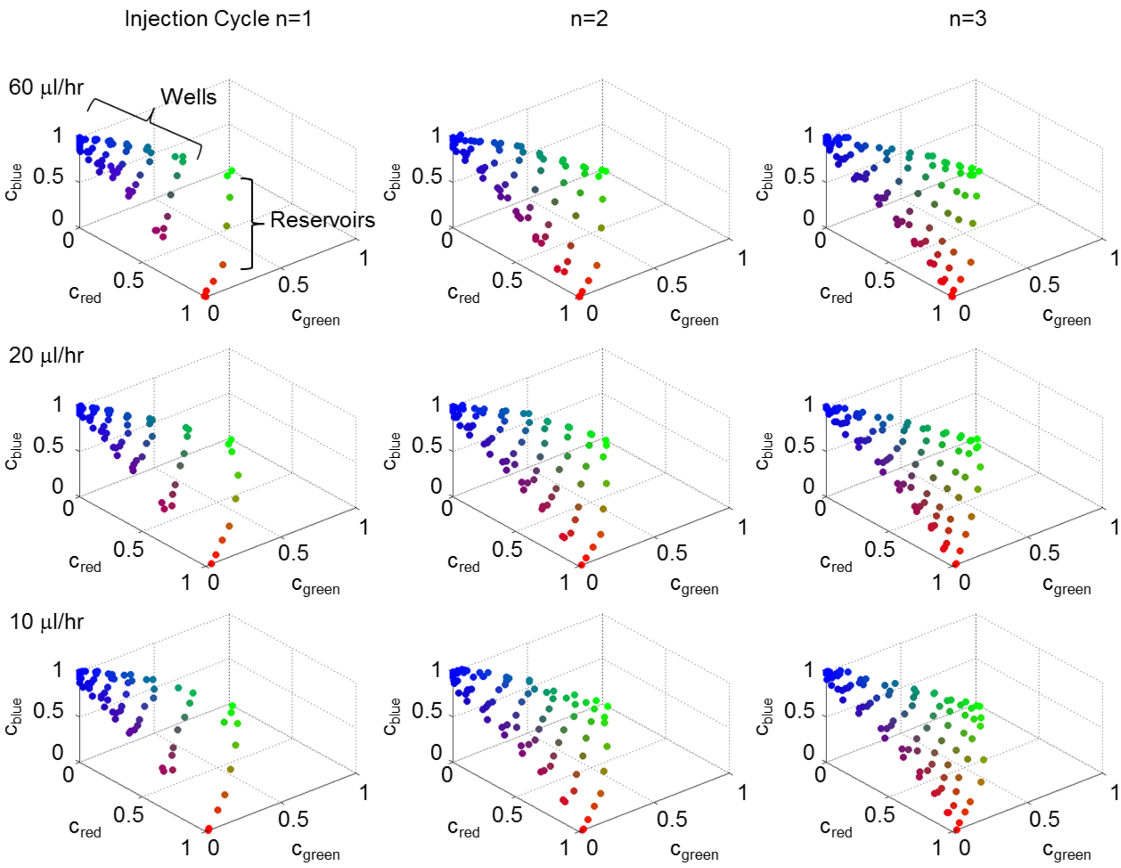



**Figure S10. Relative distribution of the three device inputs in the reservoirs and wells, for additional TLGG input flow rates.** For general details see the caption to Figure 4 in the main text. Concentration space (c-space) plots of the fractional composition of the reservoirs and wells after injections 1,2,3 at the TLGG input flow rates of 10, 20, and 60 µL $h^{-1}$. $c_{green}$, $c_{red}$, $c_{blue}$ are the normalized concentrations of the TLGG inputs 1, 2 and the pre-fill solution, respectively; each point represents the fractional composition of a particular well. The points lie in the plane $x + y + z = 1$ since the fractions add to 1. Norm. conc. = normalized concentration.

## Supplementary *Mathematica* script

Derivation and solution of recursive equations. Best fit of parameters. Plotting solutions.

## model and derivation and functions

```
recursiveformula = {c[m_, n_] :→ If[n > 0, f v co[m - 1, n] + (1 - f v) c[m, n - 1], c0],
   co[m_, n_] :→ If[m > 0, (1 - f) co[m - 1, n] + f c[m, n - 1], c1]};
```

$$\text{coeff} = \text{an}[A\_, B\_, k\_, n\_] :\to \sum_{j=0}^{k-2} A^{k-2-j} B^{j} \frac{\text{Binomial}[j+n, n]\ \text{Binomial}[k-2, j]}{j+1};$$

$$\text{solution} = \Big\{\text{co}[1, n\_] \to c1 + (c0 - c1)\ f\ (1 - f\ v)^{n-1},\ c[m\_, n\_] :\to \text{If}\Big[m > 1,$$
$$c[1, n] + (c0 - c1)\ f^2\ v\ n \sum_{k=2}^{m} \left((1 - f\ v)^{n-k+1}\ \text{an}\left[1 - f\ v - f,\ f^2\ v,\ k,\ n\right]\right),\ c1 + (c0 - c1)\ (1 - f\ v)^{n}\Big]\Big\};$$

```
model[mlist_, n_, cc0_, cc1_, vv_, ff_] := Map[c[#, n] &, mlist] //. solution /. coeff /.
   {c0 → cc0, c1 → cc1, v → vv, f → ff};
BestFit[f_, dat_, c0_, c1_, v_] := Module[{ertot, nn},
  ertot = 0;
  For[nn = 1, nn ≤ Length[dat], nn++,
   ertot = ertot + Total[(model[Range[1, Length[dat[[nn]]]], nn, c0, c1, v, f] - dat[[nn]])^2];
  ];
```

$$\frac{\sqrt{\text{ertot}}}{\text{Total}[\text{Total}[\text{dat}]]}$$

```
 ]
Print["Define marker geometries"];
<< Polytopes`
```

$$\text{v1} = \left\{\left\{-\frac{1}{2}, -\frac{\sqrt{3}}{4}\right\}, \left\{0, \frac{\sqrt{3}}{4}\right\}, \left\{\frac{1}{2}, -\frac{\sqrt{3}}{4}\right\}, \left\{-\frac{1}{2}, -\frac{\sqrt{3}}{4}\right\}\right\};$$

$$\text{v2} = \left\{\left\{-\frac{1}{2}, -\frac{1}{2}\right\}, \left\{-\frac{1}{2}, \frac{1}{2}\right\}, \left\{\frac{1}{2}, \frac{1}{2}\right\}, \left\{\frac{1}{2}, -\frac{1}{2}\right\}, \left\{-\frac{1}{2}, -\frac{1}{2}\right\}\right\};$$

```
v3 = Vertices[Pentagon];
{m1, m2, m3, m4} =
  Graphics /@ {Circle[{0, 0}, 1], Line[v1], Line[v2], Line[Join[v3, {v3[[1]]}]]};
{g1, g2, g3, g4, g4b} = Graphics /@ {{GrayLevel[0.7], Disk[{0, 0}, 1]},
    {GrayLevel[0.7], Polygon[v1]}, {GrayLevel[0.7], Polygon[v2]}, {GrayLevel[0.7],
     Polygon[Join[v3, {v3[[1]]}]]}, {GrayLevel[0.4], Polygon[Join[v3, {v3[[1]]}]]}};
{b1, b2, b3, b4} = Graphics /@ {Disk[{0, 0}, 1], Polygon[v1],
    Polygon[v2], Polygon[Join[v3, {v3[[1]]}]]};
tlist = Range[1, 3];
msz = 0.05;
plotprofile[ff_, vv_, nwells_] := Module[{pllist, pllistpos, array},
   array[n_] := model[Range[1, nwells], n, 0, 1, vv, ff];
   pllist = Map[array[#] &, tlist];
   pllistpos = Map[f[#] &, pllist] /. f[a_] :→ Select[a, # > 0.001 &];
   Print[ff];
   For[ii = 1, ii ≤ Length[pllistpos], ii++, Print[pllistpos[[ii]]]];
   ListPlot[pllistpos, PlotMarkers → {{m1, msz}, {m2, msz}, {m3, msz}, {m4, msz}},
    PlotRange → {{0, 10.3}, {-0.03, 1.03}}]
  ];
```

```
Define marker geometries
```

### Check model

```
Print["check solution to recursive formulas by direct substitution"];
ermax = 0;
For[mm = 1, mm ≤ 10, mm++, For[ii = 1, ii ≤ 10, ii++,
  x = Simplify[(c[mm, ii] //. solution /. coeff) - (c[mm, ii] //. recursiveformula) /.
     {v → 1.984, f → 0.3173, c1 → 0.89483, c0 → 0.29394}];
  y = If[mm > 1, 0, Simplify[(co[mm, ii] /. solution) - (co[1, ii] //. recursiveformula) /.
      {v → 1.984, f → 0.3173, c1 → 0.89483, c0 → 0.29394}]];
  If[ermax < Abs[x], ermax = Abs[x]];
  If[ermax < Abs[y], ermax = Abs[y]];
 ]
]
Print["max error"];
Print[ermax]
```

```
check solution to recursive formulas by direct substitution

max error
```

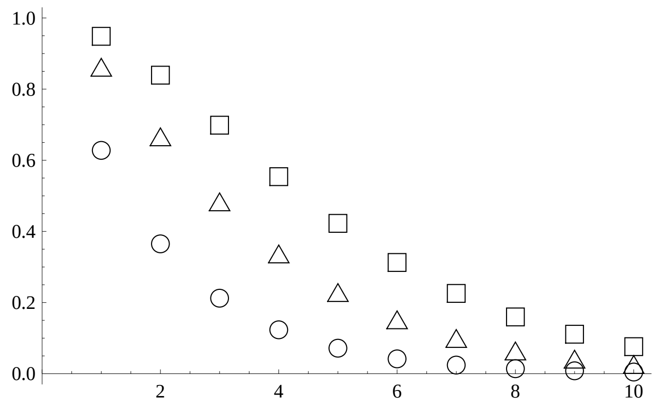
$2.22045 \times 10^{-16}$

### 10 x 10 device at 80 kPa, first 5 values for fit, first v

```
dat10b = {{0.612930343, 0.375449608, 0.254703528, 0.157277334, 0.083040116},
   {0.843750539, 0.672839057, 0.528147828, 0.387981586, 0.259374397},
   {0.939706699, 0.84131527, 0.730490865, 0.593810563, 0.452056953}};
FindMinimum[BestFit[f, dat10b, 0, 1, v], {f, 0.4}, {v, 1.5}]
FindMinimum[BestFit[f, dat10b, 0, 1, 1.5], {f, 0.4}]
plotprofile[0.41855184234674164, 1.5, 10]
```

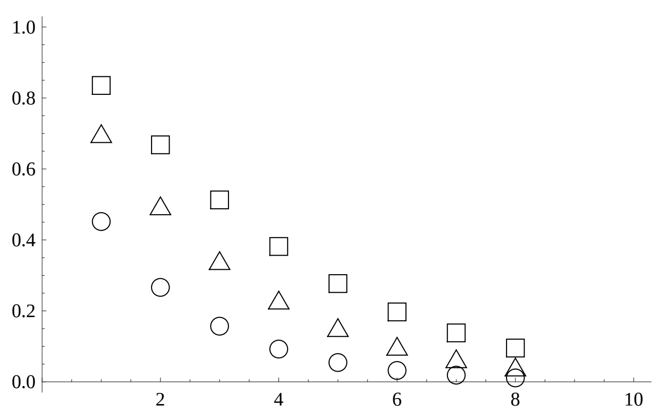
```
{0.00510537, {f → 0.372245, v → 1.65301}}

{0.014716, {f → 0.418552}}
```

```
0.418552

{0.627828, 0.365049, 0.212257, 0.123417, 0.0717603,
 0.0417249, 0.0242609, 0.0141064, 0.00820217, 0.00476913}

{0.861488, 0.66589, 0.483108, 0.336679, 0.228192,
 0.151539, 0.0990766, 0.0639831, 0.0409097, 0.0259422}

{0.94845, 0.839256, 0.698667, 0.553947, 0.422633, 0.312722, 0.225733, 0.15966, 0.111027, 0.0761088}
```




- **8 x 8 device at 20 kPa, based on first 5 values, first v**

```
dat88p20b = {{0.477992636, 0.307404563, 0.201649348, 0.112628565, 0.023451611},
{0.68308727, 0.513743802, 0.386184403, 0.261115232, 0.177853245},
{0.813873709, 0.661347069, 0.526701966, 0.388395767, 0.281418575}
  };
FindMinimum[BestFit[f, dat88p20b, 0, 1, v], {f, 0.4}, {v, 1}]
FindMinimum[BestFit[f, dat88p20b, 0, 1, 1.1], {f, 0.4}]
plotprofile[0.41070628371381857, 1.1, 8]

{0.0142326, {f → 0.382701, v → 1.17339}}

{0.017732, {f → 0.410706}}

0.410706

{0.451777, 0.266229, 0.156887, 0.0924527, 0.0544818, 0.0321058, 0.0189197, 0.0111493}

{0.699451, 0.496008, 0.341693, 0.230468, 0.152967, 0.100252, 0.0650349, 0.0418351}

{0.835232, 0.667934, 0.512724, 0.381504, 0.276985, 0.19715, 0.138047, 0.0953417}
```



Electronic Supplementary Material (ESI) for Lab on a Chip
This journal is © The Royal Society of Chemistry 2011



### 10 x 10 device, each column, first 5 wells, fixed v = 1.5

```
ColMinF[dat1_, dat2_, dat3_, vv_] := Module[{nn, xx, fm, vm, flist, vlist},
   flist = Table[0, {i, Length[dat1]}]; vlist = flist;
   If[vv == 0, Print["{column #, best fit f, best fit v}"],
    Print["{column #, best fit f, v fixed}"]];
   For[nn = 1, nn ≤ Length[dat1], nn++,
    If[vv == 0,
     xx = FindMinimum[BestFit[f, {dat1[[nn]], dat2[[nn]], dat3[[nn]]}, 0, 1, v],
       {f, 0.4, 0, 1}, {v, 1.5, 1, 2}];
     fm = xx /. {aa_, {f → fff_, v → vvv_}} → fff; flist[[nn]] = fm;
     vm = xx /. {aa_, {f → fff_, v → vvv_}} → vvv; vlist[[nn]] = vm, xx = FindMinimum[
       BestFit[f, {dat1[[nn]], dat2[[nn]], dat3[[nn]]}, 0, 1, vv], {f, 0.4, 0, 1}];
     fm = xx /. {aa_, {f → fff_}} → fff; flist[[nn]] = fm;
     vm = vv;
    ];
    Print[{nn, fm, vm}];
   ];
   If[vv == 0, Print["Mean and std of vlist ", {Mean[vlist], StandardDeviation[vlist]}]];
   Print["Mean and std of flist ", {Mean[flist], StandardDeviation[flist]}];
  ];
dat1 = {{0.563045391, 0.408317385, 0.25689542, 0.155931973, 0.108943169},
   {0.618941298, 0.386054531, 0.252603121, 0.156804711, 0.086707435},
   {0.641229226, 0.414296586, 0.279324405, 0.1885457, 0.109446159},
   {0.605234807, 0.405274429, 0.292113967, 0.182130081, 0.105567549},
   {0.584603899, 0.387150256, 0.272123276, 0.162764532, 0.08311727},
   {0.597984089, 0.34953916, 0.230821509, 0.136837889, 0.070834859},
   {0.592086738, 0.343518024, 0.224653848, 0.145031407, 0.067061654},
   {0.646907398, 0.342905228, 0.238489951, 0.140379081, 0.069276037},
   {0.59433596, 0.362962303, 0.237405343, 0.143338061, 0.068596212},
   {0.635269891, 0.367707093, 0.250145921, 0.154113242, 0.072955204}};
dat2 = {{0.803867439, 0.672773822, 0.516103764, 0.384351642, 0.284720161},
   {0.848864572, 0.677075693, 0.53362383, 0.399896786, 0.279477292},
   {0.872199155, 0.719335767, 0.575316277, 0.449639584, 0.317319527},
   {0.845433146, 0.696436184, 0.574579979, 0.424545708, 0.3037339},
   {0.824702983, 0.676058105, 0.55227412, 0.396842339, 0.272858602},
   {0.831373293, 0.644829223, 0.493929225, 0.361030596, 0.236439786},
   {0.826401467, 0.641726756, 0.491742375, 0.370321435, 0.226236322},
   {0.869156592, 0.660298881, 0.513340164, 0.365507707, 0.242813976},
   {0.833388943, 0.669395045, 0.503884033, 0.355308587, 0.224290138},
   {0.8602705, 0.668820145, 0.504815322, 0.345890523, 0.198371446}
```

Electronic Supplementary Material (ESI) for Lab on a Chip
This journal is © The Royal Society of Chemistry 2011



```
};
dat3 = {{0.914613405, 0.831588829, 0.716370447, 0.595832383, 0.488320525},
{0.943053283, 0.839049511, 0.736281438, 0.610487274, 0.471154414},
{0.95941246, 0.876462312, 0.773086293, 0.658768377, 0.523038495},
{0.934353933, 0.85523607, 0.765786403, 0.62538219, 0.500445207},
{0.919000748, 0.834621952, 0.74529576, 0.593478039, 0.460536326},
{0.920314439, 0.803819389, 0.682094999, 0.553799801, 0.410951444},
{0.923499145, 0.806237066, 0.687971837, 0.571384776, 0.411596989},
{0.965515304, 0.844325017, 0.724720065, 0.576193214, 0.436516876},
{0.969954691, 0.884716122, 0.74626346, 0.575694987, 0.420892173},
{0.947824884, 0.843345003, 0.710622714, 0.54737494, 0.378938618}
};
Print["run routines to find best fit f values"];
Print["best fit on repetition and column averaged data for f,v"];
FindMinimum[BestFit[f, dat10b, 0, 1, v], {f, 0.4, 0, 1}, {v, 1.5, 1, 2}]
Print["best fit on repetition and column averaged data for f, v=1.5"];
FindMinimum[BestFit[f, dat10b, 0, 1, 1.5], {f, 0.4, 0, 1}]
Print["best fit on repetition data for f,v"];
ColMinF[dat1, dat2, dat3, 0]
Print["best fit on repetition data for f, v=1.5"];
ColMinF[dat1, dat2, dat3, 1.5]
```

```
run routines to find best fit f values

best fit on repetition and column averaged data for f,v

{0.00510537, {f → 0.372245, v → 1.65301}}

best fit on repetition and column averaged data for f, v=1.5

{0.014716, {f → 0.418552}}
```

```
best fit on repetition data for f,v

{column ⌗, best fit f, best fit v}

{1, 0.332029, 1.73951}

{2, 0.364322, 1.69921}

{3, 0.344381, 1.88268}

{4, 0.334672, 1.84359}

{5, 0.348404, 1.71756}

{6, 0.386266, 1.52673}

{7, 0.38393, 1.53442}

{8, 0.408284, 1.56128}

{9, 0.403056, 1.55881}

{10, 0.421945, 1.49639}

Mean and std of vlist {1.65602, 0.139279}

Mean and std of flist {0.372729, 0.0324649}

best fit on repetition data for f, v=1.5

{column ⌗, best fit f, v fixed}

{1, 0.397499, 1.5}

{2, 0.422873, 1.5}

{3, 0.447894, 1.5}

{4, 0.42699, 1.5}

{5, 0.410309, 1.5}

{6, 0.394805, 1.5}

{7, 0.394889, 1.5}

{8, 0.4285, 1.5}

{9, 0.422534, 1.5}

{10, 0.420703, 1.5}

Mean and std of flist {0.4167, 0.0172271}
```

```
Show[ListPlot[{dat1[[3]], dat2[[3]], dat3[[3]]}],
 plotprofile[0.4363724547466713, 1.6207722874757995]]
Show[ListPlot[{dat1[[3]], dat2[[3]], dat3[[3]]}], plotprofile[0.48517867883691246, 1.5]]
```