YMTHE, Volume *25*

Supplemental Information

A Triple-Fluorophore-Labeled Nucleic Acid pH Nanosensor to Investigate Non-viral Gene Delivery

David R. Wilson, Denis Routkevitch, Yuan Rui, Arman Mosenia, Karl J. Wahlin, Alfredo Quinones-Hinojosa, Donald J. Zack, and Jordan J. Green

SUPPLEMENTAL INFORMATION

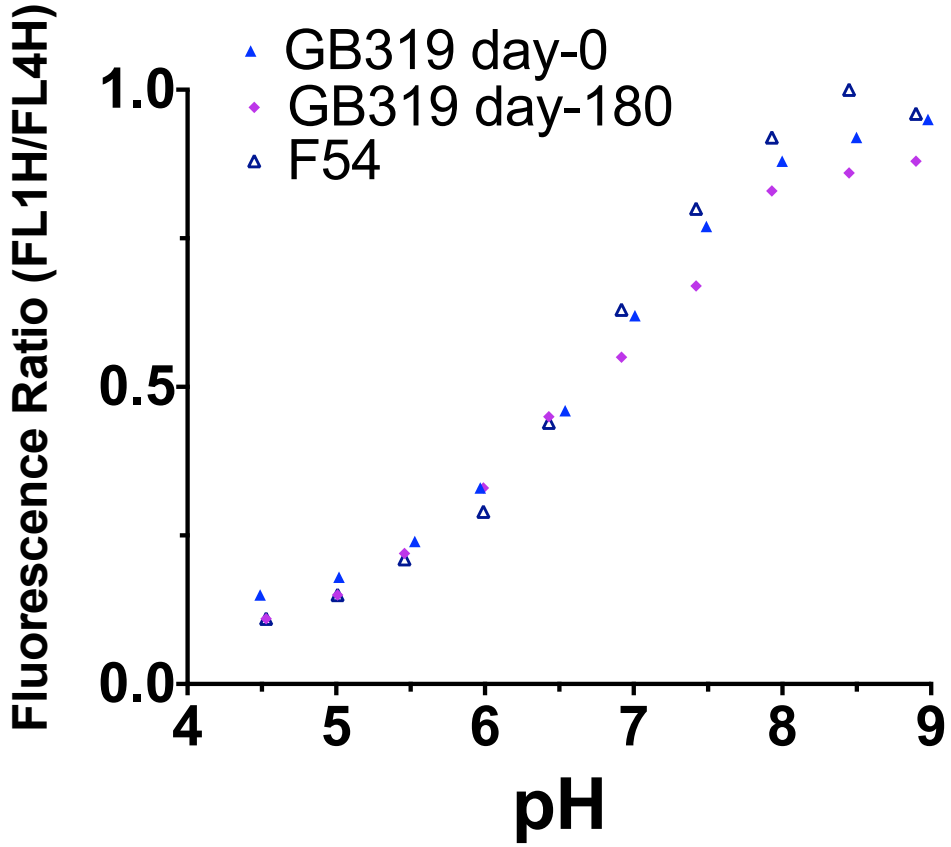
A Triple-Fluorophore Labeled Nucleic Acid pH Nanosensor to Investigate Non-Viral Gene Delivery

David R. Wilson[1-3], Denis Routkevitch[1,2], Yuan Rui[1,2], Arman Mosenia[2,4], Karl J. Wahlin[5,†], Alfredo Quinones-Hinojosa[6,7,‡], Donald J. Zack[5,8-10], Jordan J. Green[1-7,*]

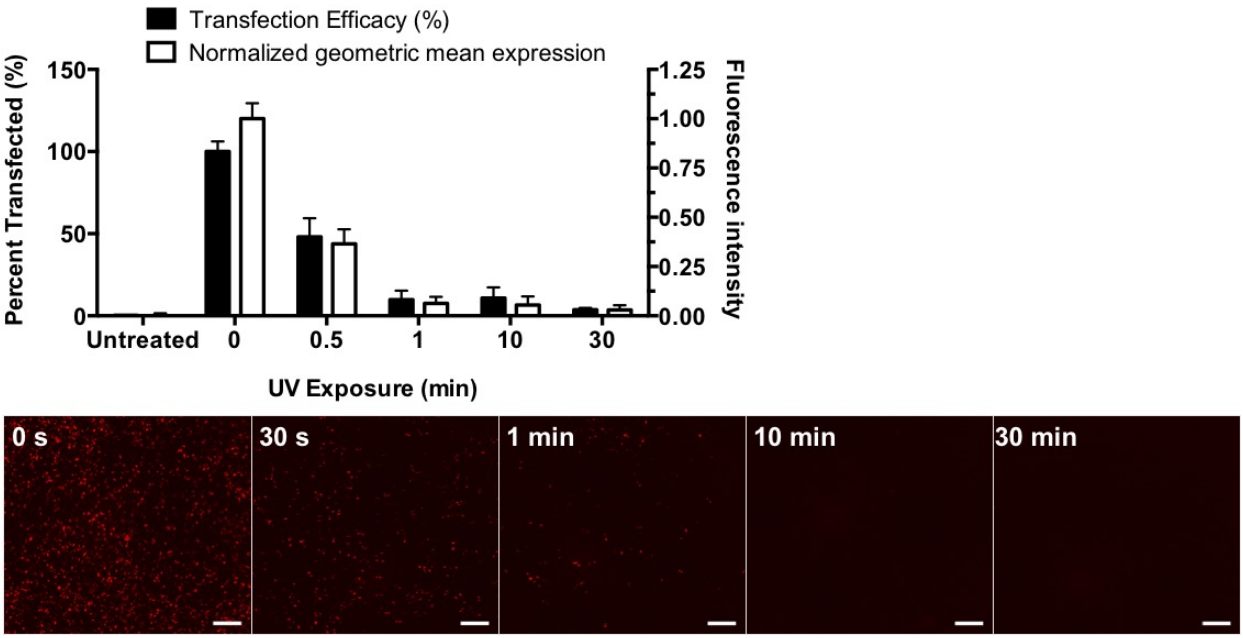
[1]Biomedical Engineering, [2]Translational Tissue Engineering Center, [3]Institute for Nanobiotechnology, [4]Materials Science and Engineering, [5]Ophthalmology, [6]Neurosurgery, [7]Oncology, [8]Neuroscience, [9]Molecular Biology and Genetics, [10]Institute of Genetic Medicine, Johns Hopkins University School of Medicine, Baltimore, MD 21231, USA
[†]Current Institution: Shiley Eye Institute, University of California, San Diego, CA 92093, USA
[‡]Current Institution: Neurologic Surgery, Mayo Clinic, Jacksonville, FL 32224, USA

*Address correspondence to:
Jordan J. Green
green@jhu.edu
400 N. Broadway
Smith Building 5017
Baltimore, MD 21231
Tel: 410-614-9113
Fax: 443-287-6298

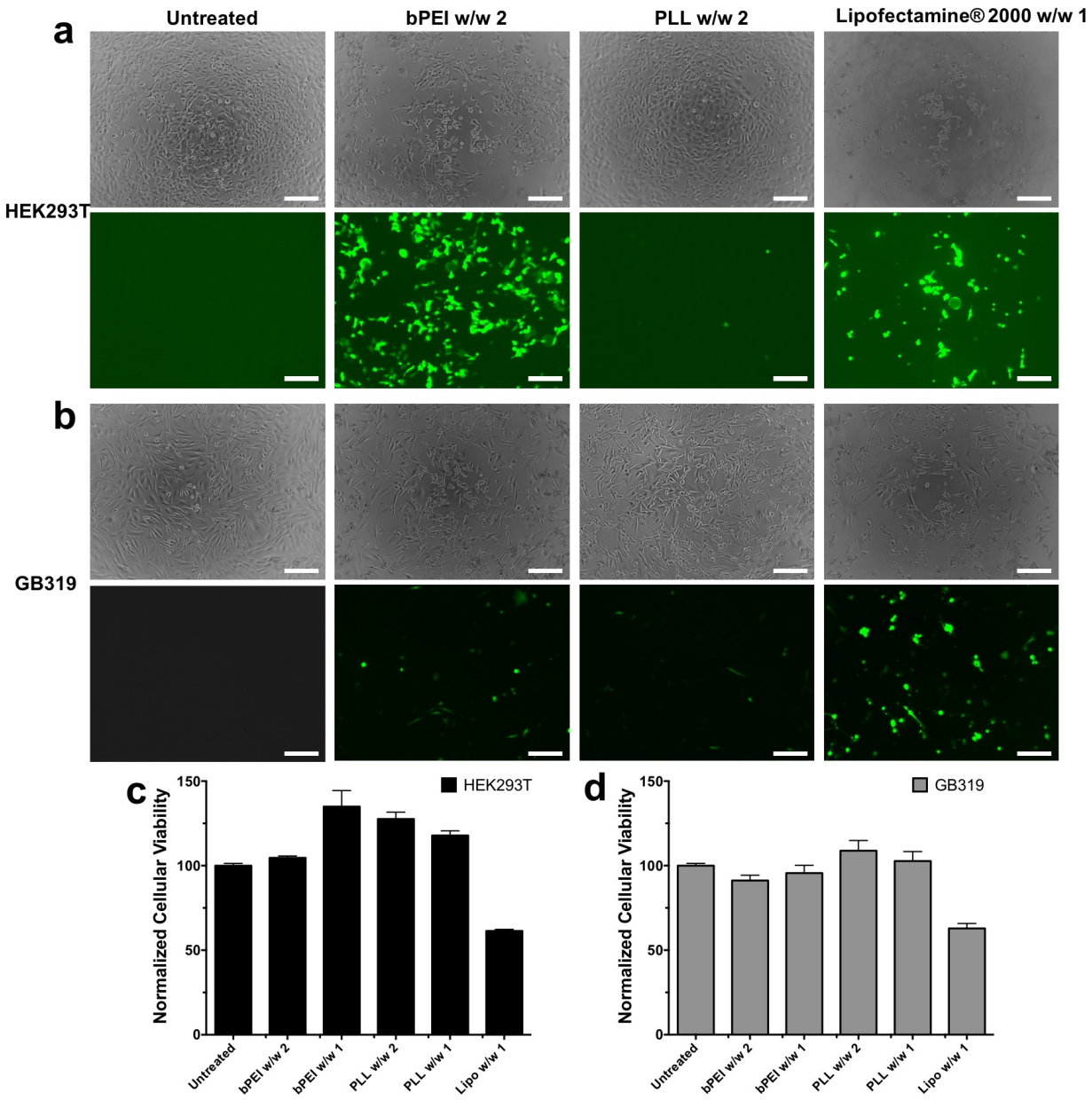
Supplementary:



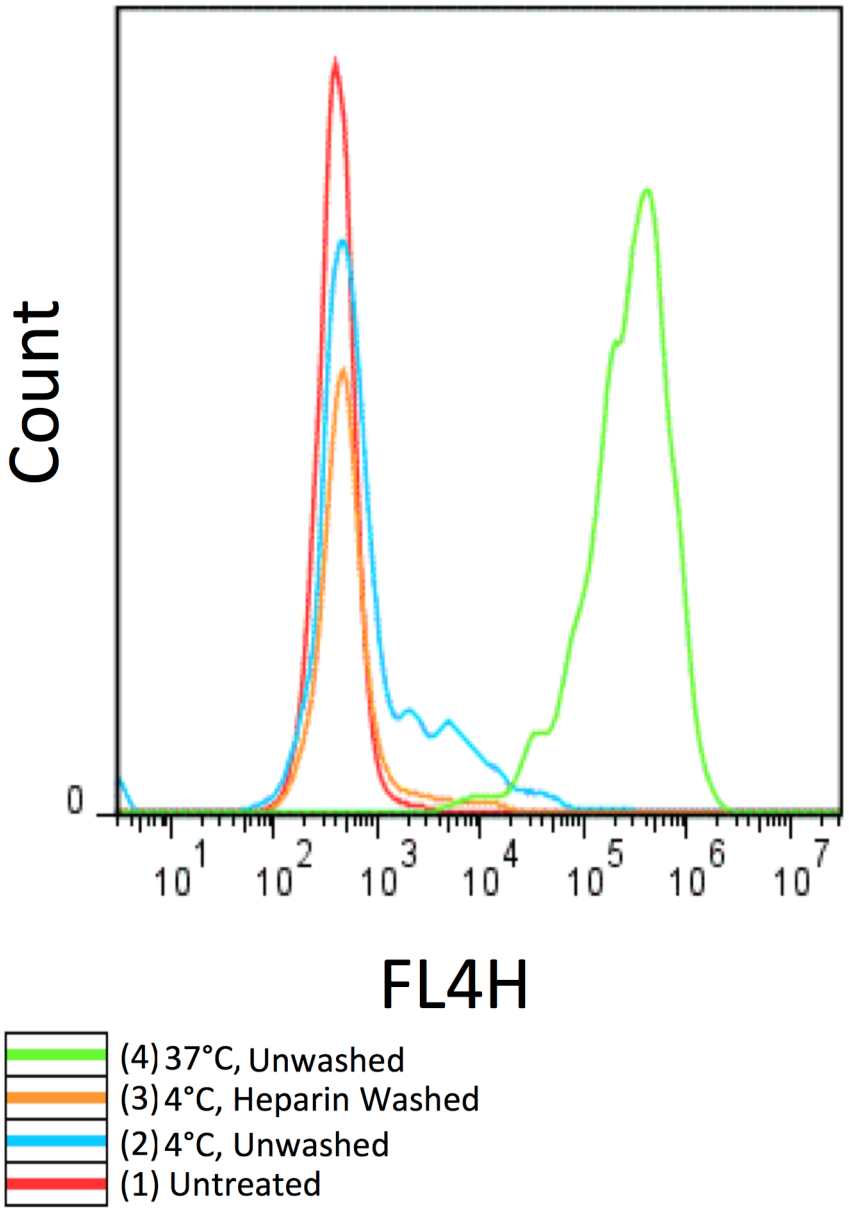
Supplementary Figure S1. **pH nanosensor flow cytometry standard curve relating pH to fluorescence ratio in two cell types.** The pH nanosensor gave consistent readings following electroporation into different cell types (GB319 and F54) and at time points six months apart.



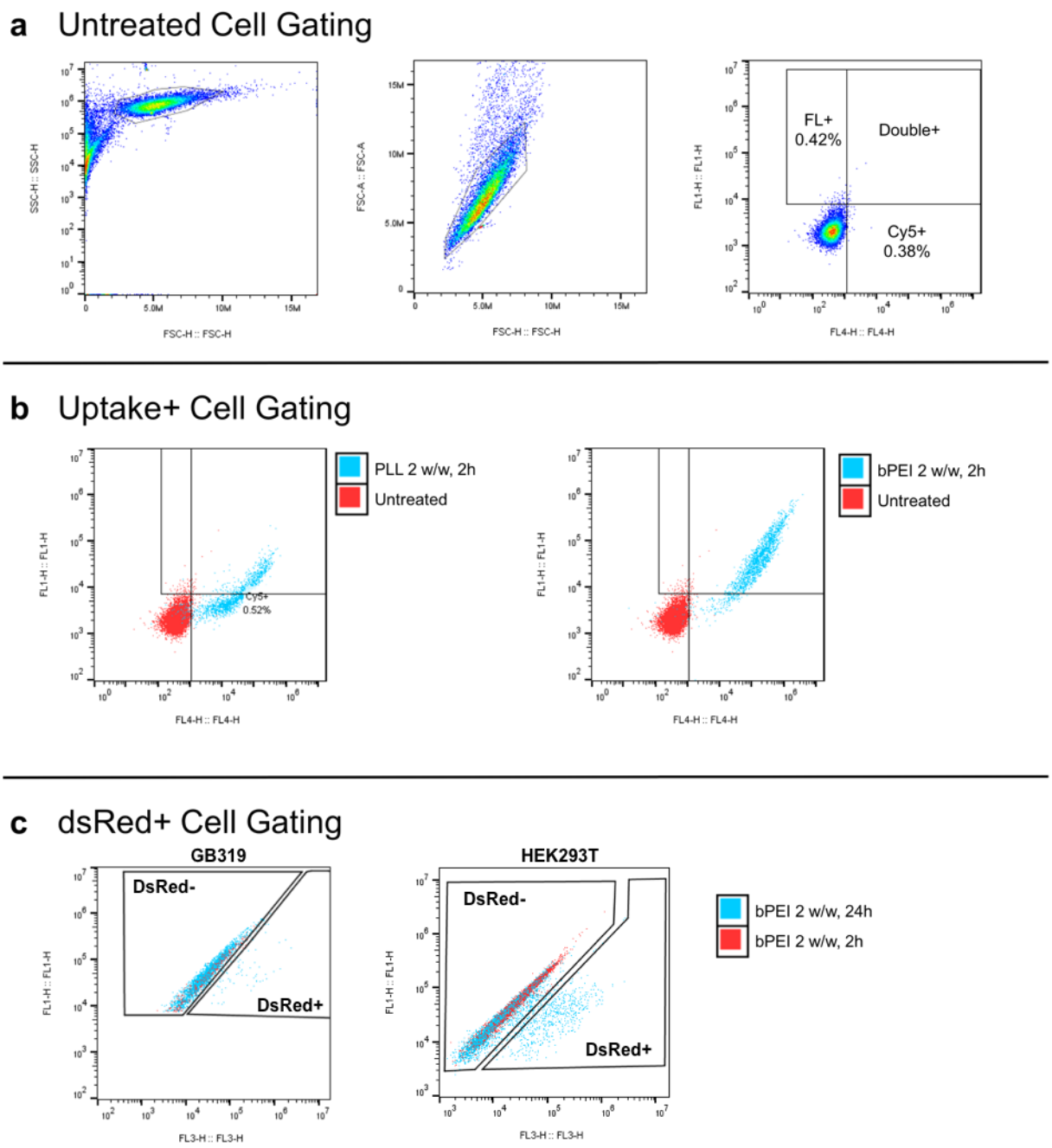
Supplementary Figure S2. UV effect on plasmid DNA expression. UV exposure resulted in decreased expression of plasmid DNA, making the labeled plasmid unsuitable by itself to induce expression. HEK293T cells were transfected with bPEI 2 w/w nanoparticles with DNA post-exposure to UV. Bars show the mean ± SEM of four wells. Transfection efficacy was assessed using flow cytometry and fluorescence intensity shows geometric mean fluorescence on the red channel normalized to that of cells transfected with un-exposed plasmid. Error bars show the mean ± SEM of four wells. Scale bars are 500 μm for 5x images.



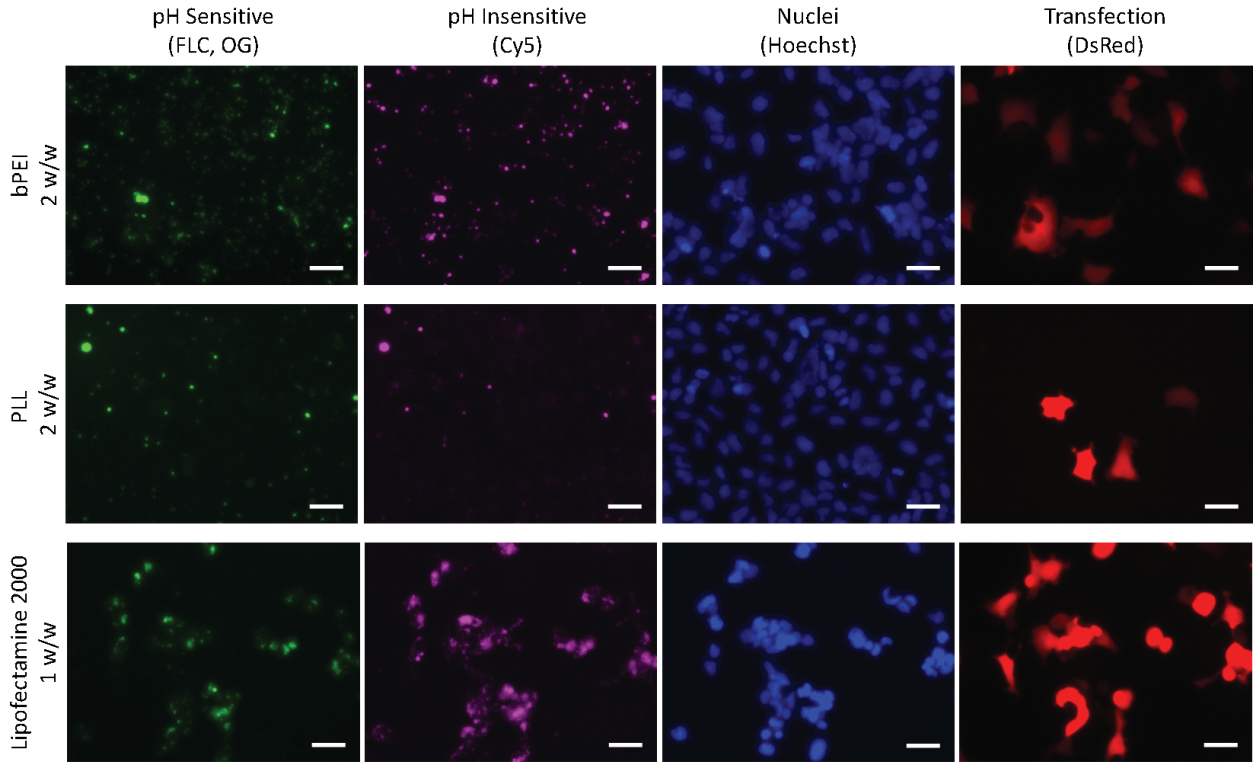
Supplementary Figure S3. Transfection efficacy in HEK293T and GB319 cells. bPEI, PLL and Lipofectamine 2000 were complexed DNA and added at a dose of 600 ng of eGFP-N1 plasmid DNA per well to determine optimal reagent dose for (**a**) HEK293T and (**b**) GB319 cells. Images were captured with a 10x fluorescence microscope with equal exposure time two days following transfection. Scale bar 200 µm. Cell viability following transfection for (**c**) HEK293T and (**d**) GB319 was assessed using MTT cell titer and was normalized to the untreated wells absorbance values. Error bars show the mean ± SEM of four wells.



Supplementary Figure S4. **Effectiveness of heparin washing.** The protocol for washing cells was confirmed to be sufficient to remove surface bound but non-internalized polyplex nanoparticles. Cells were incubated with nanoparticles for one hour at 4 °C to inhibit endocytosis then washed two times with polyanion 50 µg/mL heparin sulfate in 150 mM PBS followed by a single PBS rinse. Cells incubated at 4 °C and washed (orange) were shown to have fluorescence on the pH insensitive channel FL4H for Cy5 reduced to that of the untreated control (red) compared to cells incubated at 4°C and unwashed (blue). Cells incubated with nanoparticles under standard conditions of a two hour transfection at 37 °C (green) had over two orders of magnitude higher fluorescence than washed cells demonstrating that washing was effective to remove surface bound nanoparticles.



Supplementary Figure S5. **Flow cytometry population gating.** Flow cytometry data was gating according to the following plots. (**a**) Singlet cells were gated from all detected particles using FSC-H vs SSC-H followed by FSC-H vs FSC-A. Gates for cells with fluorescein (FL) and Cy5 fluorescence greater than the untreated population of cells were selected as shown. (**b**) Cells were gated as being positive for uptake of DNA using the Cy5 channel (FL4-H) compared to the untreated population of cells. More than 90% of cells had detectable DNA uptake for both bPEI and PLL. Cells in the Double+ region with fluorescence greater than cell autofluorescence on both channels were used for purposes of flow cytometry pH measurements. (**c**) From the cells in the Double+ region, cells were strictly gated in FL3-H vs FL1-H to determine those cells positively expressing the reporter protein dsRed at 24 hours post-transfection.



Supplementary Figure S6. HEK293T cells 24-hours post-transfection with 20% pH nanosensor DNA polyplexes show clear reporter gene expression. Microscope images were acquired with a 40x lens. Scale bar 40 µm.

Supplementary Methods; *Processing of flow cytometry data.* Flow cytometry data acquired with an Accuri C6 flow cytometer and attached HyperCyt CFlow Automator were exported to FCS files for each well. The FCS files were imported to FlowJo and analyzed as shown (Supplementary Figure S5) to identify singlet cells as well as DsRed positive and negative cell populations at 24 hours post-transfection. The individual cell data was then exported to *.csv* files from FlowJo and imported into Matlab for quantitive analysis with the following scripts. The *plateTF* script reformats vector matrices to a 96 well block plate format, while the *import_flow_data* script allows for pH calculation of individual cells as well as calculation of Pearson's correlation coefficient (PCC).

plateTF file:

```
% Plate Transform plateTF
% Serves to convert column of data for paticular variable to 96
(24x4) well plate
% format

function [plate96] = plateTF(varTF, col)
plate96 = [];
if ~exist('col') %sets default number of columns at 12
    col = 12;
end
if size(varTF,2)>size(varTF,1);
    varTF = varTF';
end
% define variable of interest to rearrange in other script:
% varTF = ____
clear plate96;
if rem(length(varTF),4) ~= 0;     % Eliminate mean/std rows if
necessary
varTF(length(varTF),:)=[];
varTF(length(varTF),:)=[];
else
end

num = ceil(length(varTF)/4);
if num <=12;
    plate96=[length(varTF(1,:))*6-2, num];
else
    plate96=[length(varTF(1,:))*10-2, 12];
end

for i0 = 0:length(varTF(1,:))-1

    num = ceil(length(varTF)/4); % determined number of columns
to plot

    if num <= 12;
        for i = 1:num; % columns
```

```
            for j = 1:4; % rows
                plate96(6*i0+j,i) = varTF(j*num-num+i, i0+1);
            end
        end

    else if num >=13
        for i = 1:col; % columns
            for j = 1:4; % rows
                plate96(10*i0+j,i) = varTF(j*col-col+i, i0+1);
            end
        end

        num = (num-col);
        for i = 1:num; % columns
            for j = 1:4; % rows
                clear j2;
                j2 = j+4;
                plate96(10*i0+j2,i) = varTF(j*num+(4*col-num)+i,
i0+1);
            end
        end
    else
        end

    end % for if/else statement
end
end
```

import_flow_data file:

```
%######### Import all Flow Cell Uptake Data #############

clear

%set(0,'DefaultFigureVisible','off');
homedir = cd;
addpath(genpath(homedir));
CSV = dir('*.csv');
CSV = {CSV.name};
AD = {}; % Array for analysis data
Correlation_Coefficients = {};

for i = 1:size(CSV,2)  % Runs for loop for total number of
files
   % Establish or reset variables
    FL1A = [];
    FL1H = [];
    FL2H = [];
```

```
    FL2A = [];
    FL3H = [];
    FL3A = [];
    FL4H = [];
    FL4A = [];

    % Import data from CSV file ending in .csv
    fid = fopen(CSV{i},'r');
    D = textscan(fid, repmat('%s',1,13), 'delimiter',',',
'CollectOutput',true);
    D = D{1};
    fclose(fid);
    sz = size(D);

    if sz(1) > 4

    % Create matrix with names of each well
    loc1 = strfind(CSV{1,i},'.csv');
     loc2 = strfind(CSV{1,i},'.');
     CSV{1,i} = CSV{1,i}(8:(loc1-1)); % Parses .exported.csv
from title

        % Run loop to read data from individual CSV file
        % Will need to adjust index values of variable D for the
column
    for j = 2:sz(1,1);

       FL1H = [FL1H; str2double(D{j,6})];
       FL2H = [FL2H; str2double(D{j,8})];
       FL3H = [FL3H; str2double(D{j,10})];
       FL4H = [FL4H; str2double(D{j,12})];

       FL1A = [FL1A; str2double(D{j,5})];
       FL2A = [FL2A; str2double(D{j,7})];
       FL3A = [FL3A; str2double(D{j,9})];
       FL4A = [FL4A; str2double(D{j,11})];
    end

        % Store data in variable analysis data "AD"
        AD{i,1} = CSV{1,i};                    % Cell Title
        AD{i,2} = length(FL1H);
        cell_count(i) = length(FL1H);
        AD{i,3} = median(FL1H);
        AD{i,4} = median(FL4H);
        AD{i,5} = median(FL1H./FL4H);          % Per cell
median ratio
        AD{i,6} = mean(FL1H);
        AD{i,7} = mean(FL4H);
```

```
        AD{i,8} = mean(FL1H./FL4H);

      Ratio = FL1H./FL4H; % Matrix with individual cell ratio
values
      FL1HT{i} = FL1H;  % Array with all FL1H./FL4H values
      FL4HT{i} = FL4H;
      FL2HT{i} = FL2H;
      FL3HT{i} = FL3H;
      FL3AR{i} = FL3A;
      FL2AR{i} = FL2A;
      FL4AR{i} = FL4A;
      FL3div1AR{i,1} = FL3H./FL1H; % Required due to bleed over
fluorescence

    %%%%%%%%% Determine pH
       pHcurve = [4.493 3.848]; % Linear fit 170131
       R = AD{i,5} ;% Ratio for each well = median FL1H./FL4H
       pH(i,1) = pHcurve(1)*R + pHcurve(2); % computes pH for
each well
       pHT{i} =  pHcurve(1)*Ratio + pHcurve(2);
% Calculate logarithm of fluorescence area
         FL2ARL10{i,1} = log10(FL2AR{i});
         FL3ARL10{i,1} = log10(FL3AR{i});
         % FL3div1ARL10{i,1} = log10(FL3AR{i}./FL1AR
         FL4ARL10{i,1} = log10(FL4AR{i});

  % Make array for calculation correlation coefficients
  cc = [];
      cc(:,1) = pHT{i};
      cc(:,2) = FL2ARL10{i,1};
      cc(:,3) = FL3ARL10{i,1};
      cc(:,4) = FL4ARL10{i,1};
      cc(:,5) = FL3div1AR{i,1};


    for y = length(cc):-1:1;
            if cc(y,1)>9;   % Eliminate for calculated pH over 9
                cc(y,:) = [];
            elseif cc(y,4)<3  % Eliminates non-sense FL4A value
rows
                cc(y,:) = [];
            elseif cc(y,3)<3  % Eliminates non-sense FL3A value
rows
                cc(y,:) = [];
            elseif cc(y,2)<3  % Eliminates non-sense FL2A value
rows
                cc(y,:) = [];
```

```
            elseif cc(y,2)<3  % Eliminates non-sense FL1A value
rows
                cc(y,:) = [];
            elseif cc(y,4)>6.5  % Eliminates non-sense FL1A
value rows
                cc(y,:) = [];
        else
        end
    end

    Correlation_Coefficients{i} = cc;

% Calculate Pearson's Correlation Coefficients between
variables

    temp = corrcoef(cc(:,1),cc(:,2));
    PCCpH2(i) = temp(2);
    temp = corrcoef(cc(:,1),cc(:,3));
    PCCpH3(i) = temp(2);
    temp = corrcoef(cc(:,1),cc(:,4));
    PCCpH4(i) = temp(2);
    temp = corrcoef(cc(:,4),cc(:,2));
    PCC42(i) = temp(2);
    temp = corrcoef(cc(:,4),cc(:,3));
    PCC43(i) = temp(2);

    % For FL1 normalized expression fluorescence values
    temp = corrcoef(cc(:,1),cc(:,5));
    PCCpH3div1(i) = temp(2);
    temp = corrcoef(cc(:,4),cc(:,5));
    PCC4_3div1(i) = temp(2);



    CSV{1,i} % Output title for tracking purposes
    clear D;
    clear sz;
    clear title;
    end % End if statement for more than three cells analyzed
    end % Loop for collecting all data

 cell_count = plateTF(cell_count);
 pHT = pHT';

 % Rearrange Pearson's Correlation Coefficient Values to plate
format
    PCCpH2 = plateTF(PCCpH2');
    PCCpH4 = plateTF(PCCpH4');
```

```
    PCC42 = plateTF(PCC42);
    PCC43 = plateTF(PCC43);
    PCCpH3 = plateTF(PCCpH3);
    PCCpH3div1 = plateTF(PCCpH3div1);
    PCC4_3div1 = plateTF(PCC4_3div1);


% Calculate mean and median values from total pH values
for i = 1:length(pHT)
    pHmean(i) = mean(pHT{i,1});
    pHmedian(i) = median(pHT{i,1});
end
pHmean = plateTF(pHmean');
pHmedian = plateTF(pHmedian');
```