Table of Contents


```
close all; % Close all previous figures
clear all; % Clear all previous parameters
```

Background information

```
% This MATLAB script was made by iGEM TU-Eindhoven 2022. This script
 was
% used to calculate the SEAP activity from absorbance values that have
 been
% measured during a colorimetric SEAP assay.

% For this colorimetric SEAP assay we transfected HEK293T cells with
% plasmids encoding for the anti-RR120 GEMS receptor, the STAT
% transcription factor and the SEAP reporter protein. The transfected
 cells
% were treated with different concentrations of the ligand RR120.
% After 48 hours of incubation, in which the cells produced SEAP, the
 cell
% culture medium was aspirated and the colorimetric SEAP assay was
% performed. In this assay, the absorbance at 405 nm was measured
 every 30
% seconds for 1 hour.

% To calculate the SEAP activity, a calibration curve is required.
 This
% calibration curve must be made specifically for the plate reader
 that is
% used to measure the absorbance.
```

Read data

```
file = '20220701_SEAP_Assay.xlsx'; % Name of the file that contains
 the data of the assay

SEAP_data = xlsread(file, 1, 'D46:BB158'); % Read measured absorption
 values in 'file'.
                                           % Select the cells from the
 excel sheet where the absorption values are presented.
time = xlsread(file, 1, 'B46:B158'); % Read time of measurements in
 'file'
```

```
                                        % Select the cells from the excel
 sheet where the time is presented.
time = time./60; % Determine time in minutes
```

Determine maximal absorption

```
% Beer-Lambert Law: maximum absorbance is 2
% Do not proceed with measured values that exceed this maximum value
% Create a list with all indexes that excreed this value
% This way, the model only uses the absorption values up to the time
 step
% at which the maximum absorbance is exceeded.

max_abs = 2;                        % Determine maximum absorbance
idx_total = [];                     % A list with all indexes of the
 values that exceed 'max_abs'
for ii = 1:size(SEAP_data,2)        % Take all columns
    if isempty(find(SEAP_data(:,ii) > max_abs,1)) % Find values that
 exceed 'max_abs'
        % If 'max_abs' is not exceeded, the array will be empty
        % Then, 'isempty' indicates that this value is true. if-loop
        % proceeds the if statement
        idx_total = idx_total;
    else % If 'max_abs' is exceeded, 'isempty' indicates that this
 value is fale. if-loop proceeds the else statement
        idx_total = [idx_total, find(SEAP_data(:,ii) > max_abs,1)]; %
 Indexes of values higher than 'max_abs' are added to 'idx_total'
    end
end

row_idx = min(idx_total); % Find the minimal of 'idx_total' to find
 which index is the lowest and from which index the data should not be
 taken into account anymore

if isempty(row_idx) % If 'max_abs' was not exceeded, 'row_idx' is
 empty
    SEAP_data = SEAP_data;  % Then, 'SEAP_data' is not altered
    time = time;            % And 'time' is not altered
else % if 'max_abs' was exceeded
    SEAP_data = SEAP_data(1:row_idx, :);    % Change 'SEAP_data'
 upward from the first 'max_abs' value
    time = time(1:row_idx);                 % Change 'time' upward
 from the first 'max_abs' value
end

% For all samples, you want to proceed with the absorbance values for
 the
% same time range, so that the absorbance values that exceeded
 'max_abs' are not taken into account
% Therefore, the time range needs to be determined.

time_range_fit = 0:1:max(time);     % Determine the time range, and
 make a list containing every minute in this range.
```

```
act_final = [];                     % Create a list that will contain
 the calculated SEAP activity
SEAP_converted = [];                % Create a list that will contain
 the calculated values of the amount of conversion by SEAP
fitting = [];                       % Create a list to fit the data
```

Calculation

```
% To calculate the SEAP activity, a calibration curve was made that is
% described by the formula: Absorbance [AU] = a*pNP[mM]+b
% This curve is specific for the plate reader that is used to measure
 the
% absorbance.

abs_to_conc = [4.7761, 0.1616];    % Absorptivity constants a and b
 determined with the calibration curve.
vol_add = 120/5;                   % Determine volumetric scaling.
 This is specific for the protocol of the assay that has been executed
% For this assay, a total sample volume of 120 uL was used. 5 uL of
 this
% total sample volume is culture medium that was aspirated from the
 cells.


for ii = 1:size(SEAP_data,2)        % Take all the columns of
 'SEAP_data'
    temp_data = SEAP_data(:,ii);    % Temporary data: read one column
 at the time
    for jj = 1:size(temp_data, 1)   % Take a row from the column form
 'temp_data'
        temp_data(jj) = (temp_data(jj)-abs_to_conc(2))/abs_to_conc(1);
     % Convert absorption to mM pNP product using the calibration
 curve
    end


    p = polyfit(time, temp_data, 1); % Fits a first-order polynomial
 through the data points that have been calculated in temp_data

    yfit = polyval(p,time_range_fit); % Evaluate the fit of the
 polynomial through the data points within the time_range_fit


    % The slope of the polynomial determines the amount of pNPP that
 is converted to pNP by SEAP per minute.
    % The concentration of pNPP that was added to each sample was
 known.
    act = p(1)*vol_add*100000; % converts the amount of converted pNPP
 per minute to the SEAP acvitivity per L in U/L.

    act_final = [act_final, act];                   % final SEAP
 activity of 'temp_data'
```

```
    SEAP_converted = [SEAP_converted, temp_data];   % mM pNP product
 obtained by SEAP.
    fitting = [fitting, yfit'];                     % Dot product of
 yfit
end

mean_act_temp = act_final; % Temporary data of the final SEAP activity
 that will be used to calculate the mean
```

Bar plot

```
% This code was used to create a bar plot to demonstrate the SEAP
 activity
% obtained by treating the cells with different concentrations of the
% ligand RR120.


mean_act = [mean_act_temp(7), mean_act_temp(19), mean_act_temp(31);...
    mean_act_temp(8), mean_act_temp(20), mean_act_temp(32);...
    mean_act_temp(9), mean_act_temp(21), mean_act_temp(33);...
    mean_act_temp(10), mean_act_temp(22), mean_act_temp(34);...
    mean_act_temp(11), mean_act_temp(23), mean_act_temp(35);...
    mean_act_temp(12), mean_act_temp(24), mean_act_temp(16)]; % Divide
 each row into a triplet experiment

triplos_name = {'0 ng/ml', '1 ng/ml', '10 ng/ml', '30 ng/ml', '100 ng/
ml', '300 ng/ml', '1000 ng/ml'};

mean_plot = zeros(1, size(mean_act,1)); % Make an empty array with the
 same size as 'mean_act'
aa = 1; % First row
while aa <= size(mean_act,1) % For all mean_act
    mean_plot(aa) = mean(mean_act(aa,:)); % Take the mean of the
 triplet and put this value in the array
    aa = aa + 1; % Go to the next row
end

bar_width = 0.8;
CM = lines;
x_dim_figure = size(mean_act,1); % The X-dimension of the figure based
 on the amount of triplets

figure('Renderer', 'painters', 'Position', [10 10 x_dim_figure*150
 720]) % Renderer sorts the graph by order specified in the code, and
 the units wherein can be drawn
set(gcf, 'color', 'w');
bar(mean_plot, bar_width, 'Linewidth', 1, 'FaceColor', [152/255
 204/255 155/255]) % Make bar plot with mean values of the triplet
 experiments
hold on
```

```
for ii = 1:size(mean_act,1)     % For every experiment plot the mean
 values of each triplet seperate as scatter point
    bar_center = ii;            % The center of the bar is or 1, 2, 3
 or ...
    outer_dis = bar_width/5;    % The distance between the tree
 scatters is determined with respect to the bar_width
    x_coord = linspace(bar_center-outer_dis, bar_center+outer_dis,
 size(mean_act,2)) ; % Creates three evenly spaced x-coordinates for
 every bar at which the values of

                  % the triplets can be plotted at the middle of the
 bar minus outer disk and plus

                 % outer disk
    f1 = scatter(x_coord, mean_act(ii,:),
 100, 'o', 'MarkerEdgeColor', 'k',... % x-coordinates, y-coordinates,
 size of scatter, shape of scatter, color of edge
              'MarkerFaceColor', [247/255 179/255 43/255],... % color
 of fill
              'MarkerFaceAlpha', 0.6,... % transparency
              'LineWidth', 1); % Linewidth of edge
end

xlim([0.5, 7.5]) % Limits of x-axis
ylim([0 4000]) % Limits of y-axis
ylabel('SEAP activity [U/L]') % Label on y-axis
set(gca, 'FontName', 'Arial', 'FontSize', 18, 'Ticklength', [0.02,
 0.1], 'TickDir', 'in') % The font of current axis is arial, the font
 size is 18,
% the length of stripe under axis is ticklength respectively 2D or 3D,
% TickDir determines if stripe points in of out of the graph, ytick is
 where
% yticks appear
xticks([1 2 3 4 5 6 7]);
xticklabels(triplos_name);
title('SEAP assay 1-7-22, RR120 concentrations','FontSize',20);
```