Supplemental Document 2

The MATLAB codes for Example 1

The outline of this supplemental document is as follows: Section 1 introduces the simplified IL-6 ODE model presented in Huang et al, 2010 [1], and Section 2 lists the MATLAB codes for Example 1. The corresponding MATLAB *.m files and the *.mat document containing the experimental data for Example 1 can be downloaded from the Villanova website (to be determined).

1. The ODE model for the simplified IL-6 pathway used in Example 1

Model equations:

$dx_1/dt = p_1u^2R^2-p_2x_1-p_3x_1x_2+p_4x_3+p_5x_3-p_{11}x_1x_5+p_{12}x_6- p_{14}x_1x_7+ p_{15}x_8$;
$dx_2/dt = -p_3x_1x_2+p_4x_3+2p_6x_4$;
$dx_3/dt = p_3x_1x_2-p_4x_3-p_5x_3$;
$dx_4/dt = p_5x_3/2-p_6x_4$;
$dx_5/dt = p_7x_4\mathrm{Step}(x_{14}-1290)/(50.6+x_4)-p_8x_5-p_9x_1x_5+p_{10}x_6+p_{11}x_6$;
$dx_6/dt = p_9x_1x_5-p_{10}x_6-p_{11}x_6$;
$dx_7/dt = -p_{12}x_1x_7+p_{13}x_8$;
$dx_8/dt = p_{12}x_1x_7-p_{13}x_8$;
$dx_9/dt = p_{14}x_8x_{10}/(p_{15}+x_{10})-p_{16}x_9$;
$dx_{10}/dt = -p_{14}x_8x_{10}/(p_{15}+x_{10})+p_{16}x_9$;
$dx_{11}/dt = p_{11}x_6$;
$dx_{12}/dt = -2\ p_{17}\ x_9\ x_{12}^2$;
$dx_{13}/dt = p_{17}\ x_9\ x_{12}^2$;
$dx_{14}/dt = 1$;

State variables of the model and their initial values:

Name	Component	Initial value (nM)
x_1	$(\text{IL6-gp80-gp130-JAK})^*_2$	0
x_2	STAT3C	1000
x_3	$(\text{IL6-gp80-gp130-JAK})^*_2$-STAT3C	0
x_4	STAT3N^*- STAT3N^*	0
x_5	SOCS3	0
x_6	$(\text{IL6-gp80-gp130-JAK})^*_2$-SOCS3	0
x_7	SHP2	100
x_8	$(\text{IL6-gp80-gp130-JAK})^*_2$-SHP2-sum	0
x_9	Erk-PP	0
x_{10}	Erk	16468
x_{11}	$(\text{IL6-gp80-gp130-JAK})_2$	0
x_{12}	C/EBPβi	40.493
x_{13}	C/EBPβn	0
x_{14}	*t (time)*	0 second
u	IL-6	0.383 (i.e.,10 ng/ml)
R	Receptor	8

Values of the parameters

Name	Physical Interpretation	Value
p_1	Forward rate constant for Reaction #1	9.9459e-005
p_2	Backward rate constant for Reaction #1	0.002
p_3	Forward rate constant for Reaction #2	0.0038
p_4	Backward rate constant for Reaction #2	1.502
p_5	Forward rate constant for Reaction #3	0.8156
p_6	Forward rate constant for Reaction #4	7.301e-004
p_7	Maximum rate for Reaction #5	0.0230
p_8	Forward rate constant for Reaction #6	7.476e-004
p_9	Forward rate constant for Reaction #7	4.971
p_{10}	Backward rate constant for Reaction #7	0.0400
p_{11}	Forward rate constant for Reaction #8	0.0023
p_{12}	Forward rate constant for Reaction #9	3.068e-004
p_{13}	Backward rate constant for Reaction #9	5.486e-004
p_{14}	Maximum rate for Reaction #10	16.00
p_{15}	Michaelis-Menten constant for Reaction #10	5.115e+003
p_{16}	Forward rate constant for Reaction #11	1.198e-005
p_{17}	Forward rate constant for Reaction #12	1.0 e-006

Note: The reaction numbers, #*n*, are consistent with those shown in Figure 5. First order rate constants have units of 1/s and second order rate constants of [nM-1 s-1].

2. The MATLAB codes for Example 1

The following programs are provided:

1) Main_simulation.m

   Type Main_simulation in the command window to run the simulation to plot the STAT3N*-STAT3N* profile from the original model.

2) IL6_model.m

   This is the function representing the ODE model. It is called by the main functions such as Main_simulation.m and objective function (i.e., objective.m).

3) Main_IL6_Sensitivity.m

   Type Main_IL6_Sensitivity in the command window to conduct sensitivity analysis to determine the important parameters that are the most important to the dynamic of STAT3N*-STAT3N*.

4) Main_estimation.m

   Type Main_estimation in the command window to estimate the four most important parameters determined from sensitivity analysis by minimizing the difference between the model predicted STAT3N*-STAT3N* profile and the experimental data.

5) objective.m

This function is used to quantify the sum of squared errors for a set of values for the four parameters to be estimated. This function is called by Main_estimation. m.

6) plot_result.m
   This function is used to plot the model predicted STAT3N*-STAT3N* with the experimental data. It is called by Main_estimation. m.

Each of the above six functions is posted in the following subsections. The experimental data are contained in the document "data_stat3.mat" which was made by the instructor by adding noise into the simulation result. This document and all aforementioned MATLAB programs can be available from Villanova website (to be announced).

2.1 Main_simulation.m

```
clc
clear
%% initial values
x0(1) = 0; % (IL6-R)^*_2
x0(2) = 1000; % STAT3C
x0(3) = 0; % (IL6-R)^*_2 -STAT3C
x0(4) = 0; % STAT3N*- STAT3N*
x0(5) = 0; % SOCS3
x0(6) = 0; % (IL6-R)^*_2-SOCS3
x0(7) = 100; % SHP2
x0(8) = 0; % (IL6-R)^*_2-SHP2-sum
x0(9) = 0; % ERK-PP
x0(10) = 16468; % ERK
x0(11) = 0; % (IL6-gp80-gp130-JAK)2
x0(12) = 40.493; % C/EBPβi
x0(13) = 0; % C/EBPβn
x0(14) = 0; % the time in the unit of second

%% values of parameters
para(1)= 9.9459e-004;
para(2)= 0.002;
para(3) = 0.0038;
para(4) = 1.5017;
para(5)= 0.8156;
para(6) = 7.3010e-004;
para(7) = 0.023;
para(8) = 7.4756e-004;
para(9)= 4.9709;
para(10) = 0.04;
para(11)= 0.0023;
para(12) = 3.0680e-004;
para(13) = 5.4861e-004;
para(14)= 15.9999;
para(15)= 5.1147e+003;
para(16)= 1.1978e-005;
para(17) = 1.0e-006;

u= 0.383;
R = 8;
```

```
tit = {'(IL6-R)^*_2','STAT3C','(IL6-R)^*_2 -STAT3C','STAT3N*-
STAT3N*','SOCS3','(IL6-R)^*_2-SOCS3','SHP2','(IL6-R)^*_2-SHP2-
sum','ERK-PP','ERK'};
%% perform the simulation and plot each variable from the model
[t y]=ode15s(@IL6_model,[0 5*3600],x0,[],para,u,R);
for i = 1:10
    plot(t/3600,y(:,i),'linewidth',2)
    xlabel('time (hour)','fontsize',12)
    ylabel(char(tit{i}),'fontsize',12)
    title(char(tit{i}))
    pause;
    close;
end
```

2.2 IL6_model.m

```
function dxdt = IL6_model(t,x,para,u,R)
    % t - time; x - state variable; para - parameter vector; u - IL-6; R
    - receptor
    tao=1290;%The time delay
    dxdt(1,1)=+para(1)*u^2*R^2-para(2)*x(1)-
    para(3)*x(1)*x(2)+para(4)*x(3)+para(5)*x(3)-
    para(9)*x(1)*x(5)+para(10)*x(6)-para(12)*x(1)*x(7)+para(13)*x(8);
    dxdt(2,1)=-para(3)*x(1)*x(2)+para(4)*x(3)+2*para(6)*x(4);
    dxdt(3,1)=+para(3)*x(1)*x(2)-para(4)*x(3)-para(5)*x(3);
    dxdt(4,1)=+para(5)*x(3)/2-para(6)*x(4);
    dxdt(5,1)=+para(7)*x(4)*delay_f(x(14)-tao)/(50.6 + x(4))-
    para(8)*x(5)-para(9)*x(1)*x(5)+para(10)*x(6)+para(11)*x(6);
    dxdt(6,1)=+para(9)*x(1)*x(5)-para(10)*x(6)-para(11)*x(6);
    dxdt(7,1)=-para(12)*x(1)*x(7)+para(13)*x(8);
    dxdt(8,1)=+para(12)*x(1)*x(7)-para(13)*x(8);
    dxdt(9,1)= para(14)*x(8)*x(10)/(para(15)+x(10))-para(16)*x(9);
    dxdt(10,1)= -para(14)*x(8)*x(10)/(para(15)+x(10))+para(16)*x(9);
    dxdt(11,1)= +para(11)*x(6);
    dxdt(12,1) = -2*para(17)*x(9)*x(12)^2;
    dxdt(13,1) = para(17)*x(9)*x(12)^2;
    dxdt(14,1)=1;
end

function y = delay_f(x)
    if x > 0
        y=1;
    else
        y=0;
    end
end
```

2.3 Main_IL6_Sensitivity.m

```
clc
clear
%% initial value
x0(1) = 0; % (IL6-R)^*_2
x0(2) = 1000; % STAT3C
x0(3) = 0; % (IL6-R)^*_2 -STAT3C
x0(4) = 0; % STAT3N*- STAT3N*
x0(5) = 0; % SOCS3
x0(6) = 0; % (IL6-R)^*_2-SOCS3
```

```
x0(7) = 100; % SHP2
x0(8) = 0; % (IL6-R)^*_2-SHP2-sum
x0(9) = 0; % ERK-PP
x0(10) = 16468; % ERK
x0(11) = 0; % (IL6-gp80-gp130-JAK)2
x0(12) = 40.493; % C/EBPβi
x0(13) = 0; % C/EBPβn
x0(14) = 0; % the time in the unit of second
%% parameter values
para(1)= 9.9459e-004;
para(2)= 0.002;
para(3) = 0.0038;
para(4) = 1.5017;
para(5)= 0.8156;
para(6) = 7.3010e-004;
para(7) = 0.023;
para(8) = 7.4756e-004;
para(9)= 4.9709;
para(10) = 0.04;
para(11)= 0.0023;
para(12) = 3.0680e-004;
para(13) = 5.4861e-004;
para(14)= 15.9999;
para(15)= 5.1147e+003;
para(16)= 1.1978e-005;
para(17) = 1.0e-006;

u= 3.830000e-001;
R = 8;
para0 = para; % record the original parameter values
[t1 x1] = ode15s(@IL6_model,[0 5*3600],x0,[],para0,u,R);% get the
simulation from the model with original values
sample_t = 0:0.1:5; % set the time points for sampling the output
profile
STAT3_1 = interp1(t1/3600,x1(:,4),sample_t); % sample the output,
STAT3N*-STAT3N*, from the simulation result at the sampling time
pointsTATE
STAT3_1_norm = norm(STAT3_1);
ratio = [0.5 1.5]; % we change the value of each parameter by +/- 50%
sensitivity = zeros(length(para),2); % sensitivity record the
sensitvity meausre, the first column for 50% decreased value and the
other for 50% increased value
for i = 1:length(para)
    para = para0; % re-set all the parameters to their original values
before changing a single parameter
    for j = 1:2
        para(i) = para0(i)*ratio(j); % change the value of each
parameter
        [t2 x2] = ode15s(@IL6_model,[0 5*3600],x0,[],para,u,R);
        STAT3_2 = interp1(t2/3600,x2(:,4),sample_t); % sample the
output at the sampling time points when the value of parameter (i) is
changed
        STAT3_2_norm= norm(STAT3_2); % quantify the norm of the sampled
output
        sensitivity(i,j) = para0(i) * (STAT3_2_norm-
STAT3_1_norm)/STAT3_1_norm/(para(i) - para0(i)); % quantify the
sensitivity measure based upon Equation (1)
```

```
        clear t2 x2 % clear t2 and x2 for the next run of simulation
    end

end

sum_sens = abs(sensitivity(:,1)) +  abs(sensitivity(:,2)); % get the
abolute sensivitiy for +/-50% change of the
[sort_sens , para_index] = sortrows( sum_sens, -1); % rank the
sensitivity measure in an acsending order, and the indexes of the
parameters are listed in the second column
disp('Rank of sensitivity measure')
sort_sens
disp('Rank of parameters: from the most to the least important')
para_index
```

2.4 Main_estimation.m

```
clc
clear
data = load('data_stat3.mat'); % load the data from the document
data_stat3.mat
sample_t = data.sample_t; % load the experiment time points 0:0.1:5
hour
STAT3 = data.STAT3; % load the measured STAT3 data: for each time point,
triple measurements are obtained
p0 = [0.0028    0.0068    0.0087    0.0006];% p0 initial guessed values
of the four parameters
[P_estimate, fval] = fmincon(@(p) objective(p,data),p0,[],[],[],[],[0 0
0 0], [1 1 1 1]);
% [0 0 0 0] and [1 1 1 1] are the lower and upper bounds of the
parameter values
% P_estimate is the estimated value for the four parameters, fval is
the SSE
plot_result(P_estimate,data) % plot the model prediction with the
experimental data
```

2.5 objective.m

```
function SSE = objective(para_est,data)
    %% initial value
    x0(1) = 0; % (IL6-R)^*_2
    x0(2) = 1000; % STAT3C
    x0(3) = 0; % (IL6-R)^*_2 -STAT3C
    x0(4) = 0; % STAT3N*- STAT3N*
    x0(5) = 0; % SOCS3
    x0(6) = 0; % (IL6-R)^*_2-SOCS3
    x0(7) = 100; % SHP2
    x0(8) = 0; % (IL6-R)^*_2-SHP2-sum
    x0(9) = 0; % ERK-PP
    x0(10) = 16468; % ERK
    x0(11) = 0; % (IL6-gp80-gp130-JAK)2
    x0(12) = 40.493; % C/EBPβi
    x0(13) = 0; % C/EBPβn
    x0(14) = 0; % the time in the unit of second
    %% parameter values
    para(1)= para_est(1);% parameter 1 for estimation
    para(2)= 0.002;
```

```
    para(3) = para_est(2);% parameter 2 for estimation
    para(4) = 1.5017;
    para(5)= 0.8156;
    para(6) = para_est(3); % parameter 3 for estimation
    para(7) = 0.023;
    para(8) = 7.4756e-004;
    para(9)= 4.9709;
    para(10) = 0.04;
    para(11)= 0.0023;
    para(12) = para_est(4); % parameter 4 for estimation
    para(13) = 5.4861e-004;
    para(14)= 15.9999;
    para(15)= 5.1147e+003;
    para(16)= 1.1978e-005;
    para(17) = 1.0e-006;
    u= 3.830000e-001;
    R = 8;
    [t x]=ode15s(@IL6_model,[0 5*3600],x0,[],para,u,R);
    y_model = interp1(t/3600,x(:,4),data.sample_t); %sample the model
    predicted STAT3N*-STAT3N* at the time points with experimental
    measurement
    y_exp = mean(data.STAT3,2)'; % use the mean value of the three data
    sets for STATN*-STAT3N* to quantify the SSE
    SSE = sum((y_model - y_exp).^2); % quantify SSE
end
```

2.6 plot_result.m

```
function SSE = plot_result(para_est,data)
    sample_t = data.sample_t;
    STAT3 = data.STAT3;

   %% initial value
    x0(1) = 0; % (IL6-R)^*_2
    x0(2) = 1000; % STAT3C
    x0(3) = 0; % (IL6-R)^*_2 -STAT3C
    x0(4) = 0; % STAT3N*- STAT3N*
    x0(5) = 0; % SOCS3
    x0(6) = 0; % (IL6-R)^*_2-SOCS3
    x0(7) = 100; % SHP2
    x0(8) = 0; % (IL6-R)^*_2-SHP2-sum
    x0(9) = 0; % ERK-PP
    x0(10) = 16468; % ERK
    x0(11) = 0; % (IL6-gp80-gp130-JAK)2
    x0(12) = 40.493; % C/EBPβi
    x0(13) = 0; % C/EBPβn
    x0(14) = 0; % the time in the unit of second
    %% parameter values
    para(1)= para_est(1);% update the value of parameter 1 with the
    estimated value
    para(2)= 0.002;
    para(3) = para_est(2);% update the value of parameter 3 with the
    estimated value
    para(4) = 1.5017;
    para(5)= 0.8156;
```

```
    para(6) = para_est(3); % update the value of parameter 6 with the
    estimated value
    para(7) = 0.023;
    para(8) = 7.4756e-004;
    para(9)= 4.9709;
    para(10) = 0.04;
    para(11)= 0.0023;
    para(12) = para_est(4); % update the value of parameter 12 with the
    estimated value
    para(13) = 5.4861e-004;
    para(14)= 15.9999;
    para(15)= 5.1147e+003;
    para(16)= 1.1978e-005;
    para(17) = 1.0e-006;

    u= 3.830000e-001;
    R = 8;
    [t x]=ode15s(@IL6_model,[0 5*3600],x0,[],para,u,R); % get the model
    prediction of STAT3N*-STAT3N* from the model

    figure

     errorbar(sample_t,mean(STAT3,2),std(STAT3,0,2),std(STAT3,0,2),'line
     width',2); % get the errorbar profile for the experimental data
     hold on; % hold the errorbar profile in the figure. This is the
     command for plotting two profiles together in a figure
     plot(t/3600,x(:,4),'r','linewidth',2); % plot the model predicted
     STAT3N*-STAT3N* profile
    xlabel('Time (hour)','fontsize',12)
    ylabel('STAT3N*-STAT3N* (nM)','fontsize',12)
    legend('Experimental data','Model prediction')
    ylim([0 100])
    xlim([0 5])

    figure
    para(7) = 0; % block the transcripton process of SOCS3
    [t2 x2]=ode15s(@IL6_model,[0 5*3600],x0,[],para,u,R);
    plot(t/3600,x(:,4),'r','linewidth',2); % plot the STAT3N*-STAT3N*
    profile for regular IL-6 signaling pathway
    hold on;
     plot(t2/3600,x2(:,4),'linewidth',2); % plot the STAT3N*-STAT3N*
     profile for IL-6 signaling pathway with SOCS3 blocked
     legend('Normal IL-6 signaling pathway','IL6 signaling pathway with
     SOCS3 transcription blocked')
    xlabel('Time (hour)','fontsize',12)
    ylabel('STAT3N*-STAT3N* (nM)','fontsize',12)
end
```


1. Huang, Z., Y.F. Chu, and J. Hahn, *Model simplification procedure for signal transduction pathway models: An application to IL-6 signaling.* Chemical Engineering Science, 2010. **65**(6): p. 1964-1975.