Table of Contents

```
function M2M_report(m2m_result)

%UNTITLED3 Summary of this function goes here
%   Detailed explanation goes here
```

Data generation

```
mydata = m2m_result.data_gen.mydata;
mydata = M2M_purge(mydata);
errordata = m2m_result.data_gen.errordata;
statenames = m2m_result.data_gen.statenames;


for i = 1:size(mydata,1)
    rect = [20 20 1024 768];
    fh = figure('Color','w','Position',rect);
    figure(fh)
    grid on
    scatter(mydata(28,:),mydata(i,:))
    hold on
    scatter(errordata(28,:),errordata(i,:))
    title(statenames(i))
    xlabel(statenames(28))
    ylabel(statenames(i))
end
```

```
Not enough input arguments.

Error in M2M_report (line 6)
mydata = m2m_result.data_gen.mydata;
```

DNA simulation

```
MYDATA = m2m_result.data_gen.MYDATA;
ip = [1,2,3,4,5,6,7,11,12];
a = floor(size(ip,2)^(1/2));
b = ceil(size(ip,2)/a);
sum=0;
rect = [20 20 1024 768];
```

```
fig_dna = figure('Color','w','Position',rect);
figure(fig_dna)
while(sum <size(ip,2))
    for i = ip
           sum = sum + 1;
           hold on
           grid on
           yyaxis left
           subplot(b,a,sum)
           scatter(MYDATA(33,:),normdata(MYDATA(i,:)));
           legend(statenames(i))
           title('DNA duplication')
           xlabel('Time [h]')
           ylabel('Concentration (a.u.)')

           yyaxis right
           scatter(MYDATA(33,:),MYDATA(32,:),'*')
           ylabel('DNA')
           axis([0 max(MYDATA(end,:))+1 1.5 4.5])
    end
end
```

Wanderlust all combinations

if doplots % subplot layout % possible combinations of dimensions in 2d C = nchoosek(1:d,2); %WChooseK ? a = floor(size(C,1)^(1/2)); b = ceil(size(C,1)/a);

rect = [20 20 800 600]; G.fh = figure('Color','w','Position',rect);

for i = 1:size(C,1)

```
  subplot(a,b,i)
[~,dens,X,Y] = kde2d(data(:,C(i,:)));
pcolor(X,Y,dens); shading interp       % density
hold on
scatter(data(:,C(i,1)),data(:,C(i,2)),1,'w.')    % all datapoints
scatter(data(G.Opts.s,C(i,1)),data(G.Opts.s,C(i,2)),1,'rx')    % start po
plot(ywant(C(i,1),:),ywant(C(i,2),:),'r','LineWidth',3)  % path
xlabel(dimension_names{C(i,1)})
ylabel(dimension_names{C(i,2)})
```

end end

Best and worse datapoints with DNA

```
y=m2m_result.analysis.new_functions.y;
for i = 1:27
plot(y(i,:))

hold on

end
legend(name([1:27]))
```

Dual combination plot

```
for i=1
    figure(i)
    plot(y(m2m_result.analysis.two_combi.combi_store{1,i}.best(1),:))
    hold on
    plot(y(m2m_result.analysis.two_combi.combi_store{1,i}.best(2),:))
    hold on
    plot(m2m_result.analysis.two_combi.combi_store{1,i}.y_previous)
end
```

Results

```
np_struct=([]);
np_problem=cell(size(m2m_result.analysis.np_problem.BEST,2),1);

for i = 1:size(m2m_result.analysis.np_problem.BEST,2)

 dual_combination=m2m_result.analysis.two_combi.combi_store{1,i}.best;
    dual_area=m2m_result.analysis.two_combi.combi_store{1,i}.area;
    np_struct.dual_combination=dual_combination;
    np_struct.dual_area=dual_area;

    np_combination = [m2m_result.analysis.np_problem.BEST{1,i}{:,2}];
    np_area = [m2m_result.analysis.np_problem.BEST{1,i}{:,5}];
    np_struct.np_combination=np_combination;
    np_struct.np_area=np_area';

    np_problem{i}=np_struct;
end

for i =1:2
    bar_data(:,i)=np_problem{i,1}.np_area;
end
% bar(np_struct.np_area)
% bar3(np_struct.np_area)
```

Published with MATLAB® R2017a