CMCN Excercise 10.11.2022

```
clear
clc
```

Create the model

From the list of reactions above, construct a model called *SampleModel* in Matlab, which contains the following fields:

- stoichiometry matrix (mxn double matrix, SampleModel.S)

```
SampleModel.S = ...
[   -1 -1 0 0 0 0 0 0 0 0 0 1   0; % RuBP
    2 1 0   -1 0 0 0 0 0 0 0 0 0; % 3PGA
    0 1 -1 0 0 0 0 0 0 0 0 0 0; % 2PG
    0 0 0 1 -2 0 -1 -1 0 -1 0 0 0; % T3P
    0 0 0 0 1 -1 0   0 0 0 0 0 0; % FBP
    0 0 0 0 0 1 -1 0   0 0 0 0 -1 % F6P
    0 0 0 0 0 0 1 -1 0 0 0 0 0; % E4P
    0 0 0 0 0 0 0 1 -1  0 0 0 0; % SBP
    0 0 0 0 0 0 0 0 1 -1 0   0 0; % S7P
    0 0 0 0 0 0 0 0 0 1 -1 0 0; % R5P
    0 0 0 0 0 0 1 0 0 1 1 -1 0]; % PP
%r: 1   2   3   4   5   6   7   8   9   10  11  12  13
```

- short metabolite names (mx1 cell array, SampleModel.mets), for simplicity use M1 to M11

```
SampleModel.mets = cellstr(strcat('M',num2str([1:size(SampleModel.S,1)]')));
```

- full metabolite names (mx1 cell array, SampleModel.metNames), use abbreviations from table above

```
SampleModel.metNames = {'RuBP';'3PGA';'2PG';'T3P';'FBP';'F6P';'E4P';'SBP';'S7P';'R5P';'PP'};
```

- short reaction names (nx1 cell array, SampleModel.rxns), for simplicity use reaction number

```
SampleModel.rxns = cellstr(strcat('R',num2str([1:size(SampleModel.S,2)]')));
```

- long reaction names (nx1 cell array, SampleModel.rxnNames), use abbreviations from table above

```
SampleModel.rxnNames ={'RUBISCO_carboxylation';'RUBISCO_oxygenation';'PGP';...
'GAPDH';'FBA1';'FBPase';'TK1';'FBA2';'SBPase';'TK2';'PPI';'PRK';'PGI'};
```

- lower and upper bounds on reaction flux (nx1 double matrix, SampleModel.lb and nx1 double matrix, SampleModel.ub), use 1000 as upper bound, use -1000 as lower bound for reversible reactions, 0 otherwise

```
SampleModel.rev = [0 0 0 1 1 0 1 1 0 1 1 0 1]';
```

```
SampleModel.ub = ones(size(SampleModel.S,2),1)*1000;
SampleModel.lb = -ones(size(SampleModel.S,2),1)*1000;
SampleModel.lb(SampleModel.rev==0) = 0;
```

We want to solve the LP

$$\max v_{\text{Rubisco}_{\text{carboxylation}}}$$

s.t.

$$\text{Nv} = 0$$
$$\text{lb} \leq v \leq \text{ub}$$

Add a vector including the right-hand side values to your model (mx1 double matrix, SampleModel.b)

```
SampleModel.b = zeros(size(SampleModel.S,1),1);
```

Add a vector including the coefficients in the objective to your model (nx1 double matrix, SampleModel.c)

```
SampleModel.c = zeros(size(SampleModel.S,2),1);
SampleModel.c(contains(SampleModel.rxnNames,'Rubisco_carb','IgnoreCase',true)) = 1;
```

Save your model in the .mat format.

```
save('SampleModel.mat',"SampleModel")
```

Use linprog to solve the LP!

```
[Sol.x,Sol.f,Sol.stat] = linprog(-SampleModel.c,[],[],SampleModel.S,SampleModel.b,SampleModel.l
```

```
Optimal solution found.
Sol = struct with fields:
       x: [13×1 double]
       f: -500
    stat: 1
Sol = struct with fields:
       x: [13×1 double]
       f: -500
    stat: 1
Sol = struct with fields:
       x: [13×1 double]
       f: -500
    stat: 1
```

```
Sol.f = -Sol.f;
```

Which reactions carry non-zero flux at the particular optimal solution?

```
SampleModel.rxnNames(Sol.x~=0)
```

```
ans = 11×1 cell
'RUBISCO_carboxylation'
'GAPDH'
'FBA1'
'FBPase'
'TK1'
```

```
'FBA2'
'SBPase'
'TK2'
'PPI'
'PRK'
     ⋮
     ⋮
```