Calculation of uptake rates

function [uptakes] = uptakes(a,b,c1,f1,p1)

%UPTAKES is a function which calculates a vector of uptake rates for different diets by multiplying basal uptake rates by random percentages of macronutrients i.e. %carbohydrate (sugars and fibers), %fat and %protein.

%This vector of uptake rates is then used to inform each novel diet formulation (with adjusted lower bounds) prior to conducting FBA.

% rng(0,'twister');

%

% a = 5; %lower and upper limits for percentages

% b = 95;

%

% c = (b-a).\*(randi(100,100,1) + a)/100;

% c = round(c);

% f = (b-a).\*(randi(100,100,1) + a)/100;

% f = round(f);

% p = (b-a).\*(randi(100,100,1) + a)/100;

% p = round(p);

%

%generate random values for carbohydrates, fats and proteins

% c = round(randi(100,1));

% f = round(randi(100,1));

% p = round(randi(100,1));

% c1 = c(100);

% f1 = f(100);

% p1 = p(100);

% A = [c1, f1, p1];

% S = sum(A);

% c1 = [c1/S]; % divide each value by the total sum to normalize (obtain random percentages stored in an array)

% c1 = round(c1\*100);

% c1 = c1/100;

% f1 = [f1/S];

% f1 = (round(f1\*100));

% f1 = f1/100;

% p1 = [p1/S];

% p1 = (round(p1\*100));

% p1 = p1/100;

% sumok = c1 + f1 + p1 % check sum = 1.00, i.e. percentages add up to 100 exactly

% %separate array elements back into carbohydrate, fat and protein content

%

% % load('basal.mat'); %load basal uptake rates

%

carbuptake = c1\* basal(1:21)/-0.1; % multiply basal rates for relevant macronutrients by their respective random percentages

fatuptake = f1\* basal(22:33)\*2/-0.1; % fats and protein rates are multiplied by 2 again to scale up to 100%

protuptake = p1\* basal(34:53)\*2/-0.1; % divide by -0.1 to rescale uptake values(since metabolites are taken up, values should be negative)

alluptakes = [carbuptake; fatuptake; protuptake]; % recombine into single vector of uptakes

% For each iteration, incorporate new uptake rates to simulate a new diet

% We start by creating indices for all macronutrient exchanges

ix\_fru = find(ismember(modelJoint.rxnNames,'EX\_fru[u]')==1);

ix\_glcD = find(ismember(modelJoint.rxnNames,'EX\_glc-D[u]')==1);

ix\_gal = find(ismember(modelJoint.rxnNames,'EX\_gal[u]')==1);

ix\_man = find(ismember(modelJoint.rxnNames,'EX\_man[u]')==1);

ix\_lcts = find(ismember(modelJoint.rxnNames,'EX\_lcts[u]')==1);

ix\_malt = find(ismember(modelJoint.rxnNames,'EX\_malt[u]')==1);

ix\_sucr = find(ismember(modelJoint.rxnNames,'EX\_sucr[u]')==1);

ix\_melib = find(ismember(modelJoint.rxnNames,'EX\_melib[u]')==1);

ix\_strch1 = find(ismember(modelJoint.rxnNames,'EX\_strch1[u]')==1);

ix\_inulin = find(ismember(modelJoint.rxnNames,'EX\_inulin[u]')==1);

ix\_levan1000 = find(ismember(modelJoint.rxnNames,'EX\_levan1000[u]')==1);

ix\_starch1200 = find(ismember(modelJoint.rxnNames,'EX\_starch1200[u]')==1);

ix\_arabinogal = find(ismember(modelJoint.rxnNames,'EX\_arabinogal[u]')==1);

ix\_pect = find(ismember(modelJoint.rxnNames,'EX\_pect[u]')==1);

ix\_pullulan1200 = find(ismember(modelJoint.rxnNames,'EX\_pullulan1200[u]')==1)

ix\_amylose300 = find(ismember(modelJoint.rxnNames,'EX\_amylose300[u]')==1);

ix\_lmn30 = find(ismember(modelJoint.rxnNames,'EX\_lmn30[u]')==1);

ix\_raffin = find(ismember(modelJoint.rxnNames,'EX\_raffin[u]')==1);

ix\_stys = find(ismember(modelJoint.rxnNames,'EX\_stys[u]')==1);

ix\_oligofru4 = find(ismember(modelJoint.rxnNames,'EX\_oligofru4[u]')==1);

ix\_dextran40 = find(ismember(modelJoint.rxnNames,'EX\_dextran40[u]')==1);

ix\_arachd = find(ismember(modelJoint.rxnNames,'EX\_arachd[u]')==1);

ix\_chsterol = find(ismember(modelJoint.rxnNames,'EX\_chsterol[u]')==1);

ix\_glyc = find(ismember(modelJoint.rxnNames,'EX\_glyc[u]')==1);

ix\_hdca = find(ismember(modelJoint.rxnNames,'EX\_hdca[u]')==1);

ix\_hdcea = find(ismember(modelJoint.rxnNames,'EX\_hdcea[u]')==1);

ix\_lnlc = find(ismember(modelJoint.rxnNames,'EX\_lnlc[u]')==1);

ix\_lnlnca = find(ismember(modelJoint.rxnNames,'EX\_lnlnca[u]')==1);

ix\_lnlncg = find(ismember(modelJoint.rxnNames,'EX\_lnlncg[u]')==1);

ix\_ocdca = find(ismember(modelJoint.rxnNames,'EX\_ocdca[u]')==1);

ix\_ocdcea = find(ismember(modelJoint.rxnNames,'EX\_ocdcea[u]')==1);

ix\_octa = find(ismember(modelJoint.rxnNames,'EX\_octa[u]')==1);

ix\_ttdca = find(ismember(modelJoint.rxnNames,'EX\_ttdca[u]')==1);

ix\_ala = find(ismember(modelJoint.rxnNames,'EX\_ala-L[u]')==1);

ix\_ser = find(ismember(modelJoint.rxnNames,'EX\_ser-L[u]')==1);

ix\_cys = find(ismember(modelJoint.rxnNames,'EX\_cys-L[u]')==1);

ix\_arg = find(ismember(modelJoint.rxnNames,'EX\_arg-L[u]')==1);

ix\_ile = find(ismember(modelJoint.rxnNames,'EX\_ile-L[u]')==1);

ix\_leu = find(ismember(modelJoint.rxnNames,'EX\_leu-L[u]')==1);

ix\_lys = find(ismember(modelJoint.rxnNames,'EX\_lys-L[u]')==1);

ix\_his = find(ismember(modelJoint.rxnNames,'EX\_his-L[u]')==1);

ix\_asn = find(ismember(modelJoint.rxnNames,'EX\_asn-L[u]')==1);

ix\_asp = find(ismember(modelJoint.rxnNames,'EX\_asp-L[u]')==1);

% ix\_thr = find(ismember(modelJoint.rxnNames,'EX\_thr-L[u]')==1);

% ix\_glu = find(ismember(modelJoint.rxnNames,'EX\_glu-L[u]')==1);

% ix\_met = find(ismember(modelJoint.rxnNames,'EX\_met-L[u]')==1);

% ix\_gln = find(ismember(modelJoint.rxnNames,'EX\_gln-L[u]')==1);

% ix\_pro = find(ismember(modelJoint.rxnNames,'EX\_pro-L[u]')==1);

% ix\_val = find(ismember(modelJoint.rxnNames,'EX\_val-L[u]')==1);

% ix\_phe = find(ismember(modelJoint.rxnNames,'EX\_phe-L[u]')==1);

% ix\_tyr = find(ismember(modelJoint.rxnNames,'EX\_tyr-L[u]')==1);

% ix\_gly = find(ismember(modelJoint.rxnNames,'EX\_gly[u]')==1);

% ix\_trp = find(ismember(modelJoint.rxnNames,'EX\_trp-L[u]')==1);

%% Set new uptake rates in the model using these indices

modelJoint.lb(ix\_fru) = alluptakes(1)

modelJoint.lb(ix\_glcD) = alluptakes(2)

modelJoint.lb(ix\_gal) = alluptakes(3)

modelJoint.lb(ix\_man) = alluptakes(4)

modelJoint.lb(ix\_lcts) = alluptakes(5)

modelJoint.lb(ix\_malt) = alluptakes(6)

modelJoint.lb(ix\_sucr) = alluptakes(7)

modelJoint.lb(ix\_melib) = alluptakes(8)

modelJoint.lb(ix\_strch1) = alluptakes(9)

modelJoint.lb(ix\_inulin) = alluptakes(10)

modelJoint.lb(ix\_levan1000) = alluptakes(11)

modelJoint.lb(ix\_starch1200) = alluptakes(12)

modelJoint.lb(ix\_arabinogal) = alluptakes(13)

modelJoint.lb(ix\_pect) = alluptakes(14)

modelJoint.lb(ix\_pullulan1200) = alluptakes(15)

modelJoint.lb(ix\_amylose300) = alluptakes(16)

modelJoint.lb(ix\_lmn30) = alluptakes(17)

modelJoint.lb(ix\_raffin) = alluptakes(18)

modelJoint.lb(ix\_stys) = alluptakes(19)

modelJoint.lb(ix\_oligofru4) = alluptakes(20)

modelJoint.lb(ix\_dextran40) = alluptakes(21)

modelJoint.lb(ix\_arachd) = alluptakes(22)

modelJoint.lb(ix\_chsterol) = alluptakes(23)

modelJoint.lb(ix\_glyc) = alluptakes(24)

modelJoint.lb(ix\_hdca) = alluptakes(25)

modelJoint.lb(ix\_hdcea) = alluptakes(26)

modelJoint.lb(ix\_lnlc) = alluptakes(27)

modelJoint.lb(ix\_lnlnca) = alluptakes(28)

modelJoint.lb(ix\_lnlncg) = alluptakes(29)

modelJoint.lb(ix\_ocdca) = alluptakes(30)

modelJoint.lb(ix\_ocdcea) = alluptakes(31)

modelJoint.lb(ix\_octa) = alluptakes(32)

modelJoint.lb(ix\_ttdca) = alluptakes(33)

modelJoint.lb(ix\_ala) = alluptakes(34)

modelJoint.lb(ix\_ser) = alluptakes(47)

modelJoint.lb(ix\_cys) = alluptakes(53)

modelJoint.lb(ix\_arg) = alluptakes(35)

modelJoint.lb(ix\_ile) = alluptakes(38)

modelJoint.lb(ix\_leu) = alluptakes(39)

modelJoint.lb(ix\_lys) = alluptakes(40)

modelJoint.lb(ix\_his) = alluptakes(45)

modelJoint.lb(ix\_asn) = alluptakes(43)

modelJoint.lb(ix\_asp) = alluptakes(44)

modelJoint.lb(ix\_thr) = alluptakes(48)

modelJoint.lb(ix\_glu) = alluptakes(36)

modelJoint.lb(ix\_met) = alluptakes(41)

modelJoint.lb(ix\_gln) = alluptakes(51)

modelJoint.lb(ix\_pro) = alluptakes(42)

modelJoint.lb(ix\_val) = alluptakes(52)

modelJoint.lb(ix\_phe) = alluptakes(46)

modelJoint.lb(ix\_tyr) = alluptakes(50)

modelJoint.lb(ix\_gly) = alluptakes(37)

modelJoint.lb(ix\_trp) = alluptakes(49)