Thermodynamically constrain a metabololic model

Author: Ronan Fleming, University of Luxembourg

Reviewers:

INTRODUCTION

In flux balance analysis of genome scale stoichiometric models of metabolism, the principal constraints are uptake or secretion rates, the steady state mass conservation assumption and reaction directionality. Von Bertylanffy [1,4] is a set of methods for (i) quantitative estimation of thermochemical parameters for metabolites and reactions using the component contribution method [3], (ii) quantitative assignment of reaction directionality in a multi-compartmental genome scale model based on an application of the second law of thermodynamics to each reaction [2], (iii) analysis of thermochemical parameters in a network context, and (iv) thermodynamically constrained flux balance analysis. The theoretical basis for each of these methods is detailed within the cited papers.

EQUIPMENT

Ubuntu 16.04 LTS

Python 2.7

NumPy 1.11.1

ChemAxon MarvinBeans 16.9.5.0

OpenBabel 2.3

EQUIPMENT SETUP

The following commands are entered in a terminal window (bash or similar shell).

Python 2

OpenBabel only works with Python 2. Most distributions should already have this installed, but if this is not the case, the following lines will do it:

sudo add-apt-repository ppa:fkrull/deadsnakes

sudo apt-get update

sudo apt-get install python2.7

NumPy

NumPy can be installed using the following commands:

sudo apt-get install python-dev

sudo apt-get install python-setuptools

sudo wget http://downloads.sourceforge.net/project/numpy/NumPy/1.11.1/numpy-1.11.1.tar.gz

sudo tar -xzvf numpy-1.11.1.tar.gz

sudo cd numpy-1.11.1

sudo python setup.py build -j 4 install

ChemAxon Calculator Plugin

ChemAxon calculator plugin requires a license. Apply for an academic license at the following link: http://www.chemaxon.com/my-chemaxon/my-academic-license/

After your license has been made available, you can download from the “My Licenses” tab on the ChemAxon website.

Download the license and place it under (replace USER by your actual user account):

/home/USER/.chemaxon

Download MarvinBeans for Linux, navigate to the directory where it was saved and make it executable (here, we downloaded version 16.9.5.0 - use the appropriate filename for your version):

sudo chmod +x marvinbeans-16.9.5.0-linux_with_jre64.sh

Execute the installer (again, use the same filename as above):

sudo ./marvinbeans-16.9.5.0-linux_with_jre64.sh

When asked for an installation directory, make it:

/opt/ChemAxon/MarvinBeans

This is important, since this is the path used by COBRA Toolbox.

Finally, add the installation path to the PATH environment variable:

PATH=$PATH:/opt/ChemAxon/MarvinBeans/bin

sudo export PATH

OpenBabel and Python bindings

Install the OpenBabel and Python 2 bindings by entering the following:

sudo apt-get install openbabel

sudo apt-get install python-openbabel

PROCEDURE

Configure the environment

With all dependencies installed correctly, we configure our environment, verfy all dependencies, and add required fields and directories to the matlab path.

```
if 1
    initVonBertalanffy
end
```

Select the model

This tutorial is tested for the E. coli model iAF1260 and the human metabolic model Recon3.0model. However, only the data for the former is provided within the COBRA Toolbox as it is used for testing von Bertylanffy, while Recon3D is not yet published and the data is not yet available. Having said this, the figures generated below are most suited to plotting results for Recon3Dmodel, so they may not be so useful for iAF1260. The Recon3Dmodel example uses values from literature for input variables where they are available.

```
if 0
    modelName='iAF1260';
else
    modelName='Recon3.0model';
end
```

Load a model

Load a model, and save it as the original model in the workspace, unless it is already loaded into the workspace.

```
clear model
if ~exist('modelOrig','var')
    switch modelName
        case 'iAF1260'
            load('iAF1260.mat');
            if model.S(952, 350)==0
                model.S(952, 350)=1; % One reaction needing mass balancing in iAF1260
            end
            model.metCharges(strcmp('asntrna[c]', model.mets))=0; % One reaction needing charg
        case 'Recon3.0model'
            modelPath='~/work/sbgCloud/programReconstruction/projects/recon2models/data/reconX
            model = loadIdentifiedModel(modelName,modelPath);
            model.csense(1:size(model.S,1),1)='E';
            %Hack for thermodynamics
            model.metFormulas{strcmp(model.mets,'h[i]')}='H';
            model.metFormulas(cellfun('isempty',model.metFormulas)) = {'R'};
            if isfield(model,'metCharge')
                model.metCharges = double(model.metCharge);
                model=rmfield(model,'metCharge');
            end
            modelOrig = model;
        otherwise
            error('setup specific parameters for your model')
    end
else
    model=modelOrig;
end
```

Set the directory containing the results

```
switch modelName
    case 'iAF1260'
        resultsPath=which('tutorial_vonBertalanffy.mlx');
        resultsPath=strrep(resultsPath,'/tutorial_vonBertalanffy.mlx','');
        resultsPath=[resultsPath filesep modelName '_results'];
        resultsBaseFileName=[resultsPath filesep modelName '_results'];
```

```
    case 'Recon3.0model'
        basePath='~/work/sbgCloud';
        resultsPath=[basePath '/programReconstruction/projects/recon2models/results/thermo/' m
        resultsBaseFileName=[resultsPath filesep modelName '_' datestr(now,30) '_'];
    otherwise
        error('setup specific parameters for your model')
end
```

Set the directory containing molfiles

```
switch modelName
    case 'iAF1260'
        molfileDir = 'iAF1260Molfiles';
    case 'Recon3.0model'
        molfileDir = [basePath '/data/molFilesDatabases/explicitHMol'];
        %molfileDir = [basePath '/programModelling/projects/atomMapping/results/molFilesDataba
        %molfileDir = [basePath '/programModelling/projects/atomMapping/results/molFilesDataba
    otherwise
        error('setup specific parameters for your model')
end
```

Set the thermochemical parameters for the model

```
switch modelName
    case 'iAF1260'
        T = 310.15; % Temperature in Kelvin
        compartments = ['c'; 'e'; 'p']; % Cell compartment identifiers
        ph = [7.7; 7.7; 7.7]; % Compartment specific pH
        is = [0.25; 0.25; 0.25]; % Compartment specific ionic strength in mol/L
        chi = [0; 90; 90]; % Compartment specific electrical potential relative to cytosol in
    case 'Recon3.0model'
        % Temperature in Kelvin
        T = 310.15;
        % Cell compartment identifiers
        compartments = ['c'; 'e'; 'g'; 'l'; 'm'; 'n'; 'r'; 'x';'i'];
        % Compartment specific pH
        ph = [7.2; 7.4; 6.35; 5.5; 8; 7.2; 7.2; 7; 7.2];
        % Compartment specific ionic strength in mol/L
        is = 0.15*ones(length(compartments),1);
        % Compartment specific electrical potential relative to cytosol in mV
        chi = [0; 30; 0; 19; -155; 0; 0; -2.303*8.3144621e-3*T*(ph(compartments == 'x') - ph(c
    otherwise
        error('setup specific parameters for your model')
end
```

Set the default range of metabolite concentrations

```
switch modelName
    case 'iAF1260'
        concMinDefault = 1e-5; % Lower bounds on metabolite concentrations in mol/L
        concMaxDefault = 0.02; % Upper bounds on metabolite concentrations in mol/L
        metBoundsFile=[];
    case 'Recon3.0model'
        concMinDefault=1e-5; % Lower bounds on metabolite concentrations in mol/L
        concMaxDefault=1e-2; % Upper bounds on metabolite concentrations in mol/L
```

```
        metBoundsFile=which('HumanCofactorConcentrations.txt');%already in the COBRA toolbox
    otherwise
        error('setup specific parameters for your model')
end
```

Set the desired confidence level for estimation of thermochemical parameters

The confidence level for estimated standard transformed reaction Gibbs energies is used to quantitatively assign reaction directionality.

```
switch modelName
    case 'iAF1260'
        confidenceLevel = 0.95;
        DrGt0_Uncertainty_Cutoff = 20; %KJ/KMol
    case 'Recon3.0model'
        confidenceLevel = 0.95;
        DrGt0_Uncertainty_Cutoff = 20; %KJ/KMol
    otherwise
        error('setup specific parameters for your model')
end
```

Prepare folder for results

```
if ~exist(resultsPath,'dir')
    mkdir(resultsPath)
end
cd(resultsPath)
```

Set the print level and decide to record a diary or not (helpful for debugging)

```
printLevel=2;
if 1
    diary([resultsPath filesep 'diary.txt'])
end
```

Setup a thermodynamically constrained model

Read in the metabolite bounds

```
if 1
    setDefaultConc=1;
    setDefaultFlux=0;
    rxnBoundsFile=[];
    model=readMetRxnBoundsFiles(model,setDefaultConc,setDefaultFlux,concMinDefault,concMaxDefa
end
```

```
Reading metabolite conc bounds from: /home/rfleming/work/sbgCloud/code/fork-cobratoolbox/src/analysis/th
              adp[c]      1e-07     0.0019
              adp[m]     0.0026     0.0094
              amp[c]      1e-07     0.0012
              atp[c]    0.00129     0.0049
              atp[m]     0.0028     0.0204
              coa[c]   2.92e-05  0.0001168
              coa[m]     0.0022     0.0039
```

```
              na1[c]      1e-07      0.025
              na1[e]     0.1326     0.1554
              nad[c] 0.00010546  0.0007572
              nad[m]     0.0005     0.0075
             nadh[c] 9.2574e-07 0.00038294
             nadh[m]      1e-07     0.0011
             nadp[c]      1e-07 5.8284e-06
             nadp[m]      1e-07     0.0015
            nadph[c]      1e-07 0.00037523
            nadph[m]      1e-07     0.0042
              nh4[c]     0.0007     0.0009
               pi[c]      0.001     0.0063
              ppi[c]     0.0021     0.0076
              udp[g]    1.4e-06    0.00014
```

Check inputs

```
model = configureSetupThermoModelInputs(model,T,compartments,ph,is,chi,concMinDefault,concMaxD
```

```
Field metCompartments is missing from model structure. Attempting to create it.
Attempt to create field metCompartments successful.

Warning: Setting temperature to a value other than 298.15 K may introduce error, since enthalpies and he
```

Check elemental balancing of metabolic reactions

```
ignoreBalancingOfSpecifiedInternalReactions=1;
if ~exist('massImbalance','var')
    if isfield(model,'Srecon')
        model.S=model.Srecon;
    end
    % Check for imbalanced reactions
    fprintf('\nChecking mass and charge balance.\n');
    %Heuristically identify exchange reactions and metabolites exclusively involved in exchang
    if ~isfield(model,'SIntMetBool')  ||  ~isfield(model,'SIntRxnBool') || ignoreBalancingOfSp
        %finds the reactions in the model which export/import from the model
        %boundary i.e. mass unbalanced reactions
        %e.g. Exchange reactions
        %     Demand reactions
        %     Sink reactions
        model = findSExRxnInd(model,[],printLevel);
    end

    if ignoreBalancingOfSpecifiedInternalReactions
        [nMet,nRxn]=size(model.S);
        ignoreBalancingMetBool=false(nMet,1);
        for m=1:nMet
%             if strcmp(model.mets{m},'Rtotal3coa[m]')
%                 pause(0.1);
%             end
            if ~isempty(model.metFormulas{m})
                ignoreBalancingMetBool(m,1)=numAtomsOfElementInFormula(model.metFormulas{m},'F
            end
        end
        ignoreBalancingRxnBool=getCorrespondingCols(model.S,ignoreBalancingMetBool,model.SIntR
        SIntRxnBool=model.SIntRxnBool;
        model.SIntRxnBool=model.SIntRxnBool & ~ignoreBalancingRxnBool;
    end

    printLevelcheckMassChargeBalance=-1;  % -1; % print problem reactions to a file
```

```
    %mass and charge balance can be checked by looking at formulas
    [massImbalance,imBalancedMass,imBalancedCharge,imBalancedRxnBool,Elements,missingFormulaeB
        = checkMassChargeBalance(model,printLevelcheckMassChargeBalance,resultsBaseFileName);
    model.balancedRxnBool=~imBalancedRxnBool;
    model.balancedMetBool=balancedMetBool;
    model.Elements=Elements;
    model.missingFormulaeBool=missingFormulaeBool;

    %reset original boolean vector
    if ignoreBalancingOfSpecifiedInternalReactions
        model.SIntRxnBool=SIntRxnBool;
    end
end
```

```
Checking mass and charge balance.
Found biomass reaction: biomass_reaction
Found biomass reaction: biomass_maintenance
Found biomass reaction: biomass_maintenance_noTrTr
ATP demand reaction is not considered an exchange reaction by default. It should be mass balanced:
DM_atp_c_ h2o[c] + atp[c]  -> h[c] + adp[c] + pi[c]
There are mass imbalanced reactions, see ~/work/sbgCloud/programReconstruction/projects/recon2models/res
There are charge imbalanced reactions, see ~/work/sbgCloud/programReconstruction/projects/recon2models/r
```

Check that the input data necessary for the component contribution method is in place

```
if ~isfield(model,'pseudoisomers')
    model = setupComponentContribution(model,molfileDir);
end
```

```
Creating MetStructures.sdf from molfiles.
Percentage of metabolites without mol files: 10.7%
Converting SDF to InChI strings.
Estimating metabolite pKa values.
Warning: File not found or permission denied
```

```
     1
10fthf
     2
10fthf5glu
     3
10fthf6glu
     4
10fthf7glu
     5
11docrtsl
     6
11docrtstrn
     7
12HPET
     8
12dgr120
     9
12harachd
    10
12htacr
    11
12ppd_R
    12
12ppd_S
    13
1331tacr
    14
```

13_cis_oretn
    15
13_cis_retn
    16
13_cis_retnglc
    17
13dampp
    18
13dmt
    19
13dpg
    20
14hmdz
    21
1513tacr
    22
1531tacr
    23
15HPET
    24
15dmt
    25
15kprostgf2
    26
17ahprgnlone
    27
17ahprgstrn
    28
18harachd
    29
1a2425thvitd2
    30
1a25dhvitd2
    31
1a25dhvitd3
    32
1hibup_S
    33
1hibupglu_S
    34
1hmdgluc
    35
1mncam
    36
1ohmdz
    37
1p3h5c
    38
1pipdn2c
    39
1pyr5c
    40
20ahchsterol
    41
21hprgnlone
    42
23cump
    43
23dh1i56dio
    44
23doguln
    45
23dpg
    46
2425dhvitd2
    47
2425dhvitd3
    48

24nph
    49
25aics
    50
25hvitd2
    51
25hvitd3
    52
2amac
    53
2aobut
    54
2c23dh56dhoxin
    55
2ddecdicoa
    56
2decdicoa
    57
2docohexecoa
    58
2docopencoa
    59
2dodtricoa
    60
2dp6mep
    61
2dp6mobq
    62
2dp6mobq_me
    63
2dpmhobq
    64
2dr1p
    65
2dr5p
    66
2h3mv
    67
2hatvacid
    68
2hatvacidgluc
    69
2hatvlac
    70
2hatvlacgluc
    71
2hb
    72
2hexdtetcoa
    73
2hexdtricoa
    74
2hibup_S
    75
2hibupglu_S
    76
2hiv
    77
2hydog
    78
2hyoxplac
    79
2kmb
    80
2m3hbu
    81
2m3hvac
    82

2m3ovac
83
2m3ovcoa
84
2maacoa
85
2mb2coa
86
2mbcoa
87
2mcit
88
2mop
89
2mp2coa
90
2mpdhl
91
2obut
92
2octdectecoa
93
2octpencoa
94
2oxoadp
95
2pg
96
31dmt
97
34dhmald
98
34dhoxmand
99
34dhoxpeg
100
34dhpac
101
34dhpe
102
34dhpha
103
34dhphe
104
34hpl
105
34hpp
106
35cgmp
107
35diotyr
108
35dsmv
109
3aib
110
3aib_D
111
3bcrn
112
3ddcrn
113
3ddecdicoa
114
3deccrn
115
3decdicoa
116

3dhguln
   117
3docopencoa
   118
3dodtricoa
   119
3dpdhb
   120
3dpdhb_me
   121
3dphb
   122
3dsphgn
   123
3h3mglt
   124
3hadicoa
   125
3hadpac
   126
3hanthrn
   127
3hbcoa
   128
3hbcoa_R
   129
3hdcoa
   130
3hddcoa
   131
3hdeccoa
   132
3hdececrn
   133
3hexdcoa
   134
3hexdcrn
   135
3hexdtetcoa
   136
3hexdtricoa
   137
3hglutcoa
   138
3hibup_S
   139
3hibutcoa
   140
3hivac
   141
3hlvst
   142
3hmbcoa
   143
3hmp
   144
3hodcoa
   145
3hpcoa
   146
3hpp
   147
3hpppn
   148
3hsmv
   149
3hsmvacid
   150

3htdcoa
151
3htmelys
152
3ityr_L
153
3ivcoa
154
3ivcrn
155
3m4hpga
156
3mb2coa
157
3mgcoa
158
3mglutac
159
3mglutr
160
3mhis
161
3mlda
162
3mldz
163
3mob
164
3mop
165
3mox4hoxm
166
3mox4hpac
167
3moxtyr
168
3mtp
169
3ocddcoa
170
3octdec2crn
171
3octdeccrn
172
3octdece1coa
173
3octdece1crn
174
3octdectecoa
175
3octpencoa
176
3odcoa
177
3oddcoa
178
3ohdcoa
179
3ohglutac
180
3ohodcoa
181
3ohsebac
182
3ohsebcoa
183
3ohsubac
184

3ohsubcoa
185
3ohxccoa
186
3otdcoa
187
3pg
188
3php
189
3sala
190
3snpyr
191
3spyr
192
3tdcrn
193
3tetd7ecoa
194
3thexddcoa
195
3ttetddcoa
196
3uib
197
3ump
198
42A3HP24DB
199
44mctr
200
44mzym
201
4aabutn
202
4abut
203
4abutn
204
4aphdob
205
4bhglz
206
4fumacac
207
4glu56dihdind
208
4h2oglt
209
4hatvacid
210
4hatvlac
211
4hbz
212
4hbzcoa
213
4hdebrisoquine
214
4hexdtetcoa
215
4hexdtricoa
216
4hglusa
217
4hmdgluc
218

4hoxpacd
   219
4hphac
   220
4hpro_LT
   221
4izp
   222
4mlacac
   223
4mop
   224
4mptnl
   225
4mtob
   226
4mtolbutamide
   227
4mzym_int1
   228
4mzym_int2
   229
4nph
   230
4nphsf
   231
4ohbut
   232
4ohmdz
   233
4ppan
   234
4ppcys
   235
4pyrdx
   236
4tmeabut
   237
4tmeabutn
   238
56dihindlcrbxlt
   239
56dthm
   240
56dura
   241
5HPET
   242
5a2opntn
   243
5adtststerone
   244
5adtststeroneglc
   245
5adtststerones
   246
5aizc
   247
5aop
   248
5cysdopa
   249
5cysgly34dhphe
   250
5dhf
   251
5dpmev
   252

5eipenc
253
5eipencoa
254
5forthf
255
5fthf
256
5g2oxpt
257
5homeprazole
258
5hoxindact
259
5hoxindoa
260
5htrp
261
5mdr1p
262
5mdru1p
263
5mta
264
5mthf
265
5ohhexa
266
5oxpro
267
5pmev
268
5tedtricoa
269
5thf
270
6a2ohxnt
271
6bhglz
272
6bhglzglc
273
6csmv
274
6csmvacid
275
6dhf
276
6hddopaqn
277
6hlvst
278
6hlvstacid
279
6hmsmv
280
6hmsmvacid
281
6hoxmelatn
282
6hsmv
283
6hsmvacid
284
6htststerone
285
6melvacid
286

6melvst
   287
6msmv
   288
6pgc
   289
6pgl
   290
6pthp
   291
6thf
   292
7bhglz
   293
7bhglzglc
   294
7dhchsterol
   295
7dhf
   296
7klitchol
   297
7ohocata
   298
7thf
   299
C01041
   300
C01601
   301
C01747
   302
C02356
   303
C02470
   304
C02528
   305
C02712
   306
C03681
   307
C04717
   308
C04805
   309
C04849
   310
C05109
   311
C05279
   312
C05280
   313
C05298
   314
C05299
   315
C05300
   316
C05301
   317
C05302
   318
C05767
   319
C05769
   320

C05770
   321
C05957
   322
C06314
   323
C06315
   324
C07297
   325
C08261
   326
C09642
   327
C10164
   328
C11695
   329
C11821
   330
C13856
   331
C14768
   332
C14769
   333
C14770
   334
C14771
   335
C14825
   336
C14826
   337
CE0074
   338
CE0233
   339
CE0347
   340
CE0693
   341
CE0713
   342
CE0737
   343
CE0785
   344
CE0849
   345
CE0955
   346
CE1243
   347
CE1261
   348
CE1273
   349
CE1297
   350
CE1310
   351
CE1352
   352
CE1401
   353
CE1447
   354

CE1556
355
CE1562
356
CE1589
357
CE1617
358
CE1918
359
CE1925
360
CE1926
361
CE1935
362
CE1936
363
CE1939
364
CE1940
365
CE1943
366
CE1944
367
CE1950
368
CE2006
369
CE2026
370
CE2028
371
CE2038
372
CE2047
373
CE2049
374
CE2053
375
CE2056
376
CE2084
377
CE2088
378
CE2089
379
CE2172
380
CE2176
381
CE2209
382
CE2211
383
CE2242
384
CE2243
385
CE2245
386
CE2246
387
CE2247
388

CE2248
389
CE2249
390
CE2250
391
CE2251
392
CE2253
393
CE2313
394
CE2314
395
CE2417
396
CE2418
397
CE2420
398
CE2421
399
CE2422
400
CE2424
401
CE2432
402
CE2433
403
CE2434
404
CE2437
405
CE2438
406
CE2439
407
CE2440
408
CE2441
409
CE2442
410
CE2445
411
CE2510
412
CE2537
413
CE2567
414
CE2576
415
CE2577
416
CE2705
417
CE2838
418
CE2839
419
CE2866
420
CE2870
421
CE2872
422

CE2873
423
CE2874
424
CE2875
425
CE2915
426
CE2916
427
CE2917
428
CE2934
429
CE3554
430
CE4633
431
CE4722
432
CE4723
433
CE4724
434
CE4790
435
CE4791
436
CE4792
437
CE4793
438
CE4794
439
CE4795
440
CE4796
441
CE4797
442
CE4798
443
CE4799
444
CE4800
445
CE4801
446
CE4802
447
CE4803
448
CE4804
449
CE4806
450
CE4807
451
CE4808
452
CE4810
453
CE4811
454
CE4812
455
CE4817
456

CE4819
457
CE4820
458
CE4821
459
CE4831
460
CE4832
461
CE4833
462
CE4834
463
CE4835
464
CE4838
465
CE4840
466
CE4841
467
CE4842
468
CE4843
469
CE4844
470
CE4845
471
CE4846
472
CE4847
473
CE4848
474
CE4849
475
CE4850
476
CE4851
477
CE4852
478
CE4853
479
CE4854
480
CE4855
481
CE4872
482
CE4874
483
CE4876
484
CE4877
485
CE4881
486
CE4888
487
CE4890
488
CE4968
489
CE4969
490

CE4970
491
CE4987
492
CE4988
493
CE4989
494
CE4990
495
CE5021
496
CE5022
497
CE5025
498
CE5026
499
CE5072
500
CE5114
501
CE5116
502
CE5117
503
CE5118
504
CE5119
505
CE5120
506
CE5125
507
CE5126
508
CE5144
509
CE5148
510
CE5150
511
CE5151
512
CE5152
513
CE5153
514
CE5154
515
CE5155
516
CE5156
517
CE5157
518
CE5158
519
CE5160
520
CE5161
521
CE5162
522
CE5166
523
CE5178
524

CE5236
525
CE5276
526
CE5304
527
CE5586
528
CE5626
529
CE5629
530
CE5643
531
CE5665
532
CE5786
533
CE5787
534
CE5788
535
CE5789
536
CE5791
537
CE5797
538
CE5798
539
CE5853
540
CE5854
541
CE5867
542
CE5868
543
CE5869
544
CE5934
545
CE5944
546
CE5945
547
CE5946
548
CE5947
549
CE5986
550
CE6031
551
CE6205
552
CE6230
553
CE6232
554
CE6234
555
CE6246
556
CE6247
557
CE6252
558

CE6504
559
CE6506
560
CE6508
561
CE7047
562
CE7079
563
CE7081
564
CE7082
565
CE7083
566
CE7085
567
CE7088
568
CE7090
569
CE7091
570
CE7096
571
CE7097
572
CE7172
573
HC00250
574
HC00319
575
HC00342
576
HC00361
577
HC00460
578
HC00576
579
HC00591
580
HC00664
581
HC00682
582
HC00718
583
HC00822
584
HC00900
585
HC00955
586
HC01104
587
HC01118
588
HC01180
589
HC01223
590
HC01254
591
HC01361
592

HC01376
593
HC01377
594
HC01397
595
HC01405
596
HC01406
597
HC01407
598
HC01408
599
HC01412
600
HC01415
601
HC01434
602
HC01440
603
HC01441
604
HC01444
605
HC01446
606
HC01459
607
HC01496
608
HC01501
609
HC01522
610
HC01577
611
HC01609
612
HC01668
613
HC01672
614
HC01700
615
HC01712
616
HC01842
617
HC02020
618
HC02021
619
HC02022
620
HC02023
621
HC02024
622
HC02025
623
HC02027
624
HC02110
625
HC02121
626

HC02180
627
HC02187
628
HC02191
629
HC02192
630
HC02193
631
HC02194
632
HC02195
633
HC02196
634
HC02197
635
HC02198
636
HC02200
637
HC02201
638
HC02202
639
HC02203
640
HC02204
641
HC02205
642
HC02206
643
HC02207
644
HC02208
645
HC02210
646
HC02213
647
HC02214
648
HC02216
649
HC02217
650
HC02220
651
HC02228
652
HC10856
653
HC10857
654
HC10858
655
HC10859
656
L2aadp
657
L2aadp6sa
658
L_dpchrm
659
Lcyst
660

```
Lcystin
   661
Lfmkynr
   662
Lhcystin
   663
Lkynr
   664
Lpipecol
   665
M00003
   666
M00004
   667
M00006
   668
M00008
   669
M00010
   670
M00011
   671
M00012
   672
M00017
   673
M00018
   674
```

Prepare the training data for the component contribution method

```
if ~exist('training_data','var')
    training_data = prepareTrainingData(model,printLevel);
end
```

```
Successfully added 3914 values from TECRDB
Successfully added 223 formation energies
Successfully added 13 redox potentials
Loading the InChIs for the training data from: /home/rfleming/work/sbgCloud/code/fork-cobratoolbox/src/a
Successfully created balanced training-data structure: 672 compounds and 4138 reactions
Loading the pKa values for the training data from: cache/kegg_pkas.mat
Warning: Estimation inaccuracy may result from missing stereo in InChI for:
1. 2hydog
2. fum
3. mescon
4. retinal
5. retinal_11_cis
6. retinal_cis_13
7. retinal_cis_9
Mapping model metabolites to nist compounds
Creating group incidence matrix
python2 /home/rfleming/work/sbgCloud/code/fork-cobratoolbox/src/analysis/thermo/componentContribution/ir
Warning: getGroupVectorFromInchi did not succeed for: InChI=1/C18H22O2/c1-18-9-8-14-13-5-3-12(19)10-11(1
python2 /home/rfleming/work/sbgCloud/code/fork-cobratoolbox/src/analysis/thermo/componentContribution/ir
Warning: getGroupVectorFromInchi did not succeed for: InChI=1/C18H24O2/c1-18-9-8-14-13-5-3-12(19)10-11(1
python2 /home/rfleming/work/sbgCloud/code/fork-cobratoolbox/src/analysis/thermo/componentContribution/ir
Warning: getGroupVectorFromInchi did not succeed for: InChI=1/C3H9NO/c1-4(2,3)5/h1-3H3
python2 /home/rfleming/work/sbgCloud/code/fork-cobratoolbox/src/analysis/thermo/componentContribution/ir
Warning: getGroupVectorFromInchi did not succeed for: InChI=1/C3H5NO2/c1-2(4)3(5)6/h1,4H2,(H,5,6)/f/h4H
python2 /home/rfleming/work/sbgCloud/code/fork-cobratoolbox/src/analysis/thermo/componentContribution/ir
Warning: getGroupVectorFromInchi did not succeed for: InChI=1/Na/q+1
python2 /home/rfleming/work/sbgCloud/code/fork-cobratoolbox/src/analysis/thermo/componentContribution/ir
Warning: getGroupVectorFromInchi did not succeed for: InChI=1/C7H7NO3/c8-6-4(7(10)11)2-1-3-5(6)9/h1-3,9H
```

```
python2 /home/rfleming/work/sbgCloud/code/fork-cobratoolbox/src/analysis/thermo/componentContribution/in
Warning: getGroupVectorFromInchi did not succeed for: InChI=1/C7H7NO5/c8-5(7(12)13)4(6(10)11)2-1-3-9/h1-
python2 /home/rfleming/work/sbgCloud/code/fork-cobratoolbox/src/analysis/thermo/componentContribution/in
Warning: getGroupVectorFromInchi did not succeed for: InChI=1/C39H49N11O18/c40-20(33(59)60)5-9-24(51)19(
python2 /home/rfleming/work/sbgCloud/code/fork-cobratoolbox/src/analysis/thermo/componentContribution/in
Warning: getGroupVectorFromInchi did not succeed for: InChI=1/C17H19N3O4S/c1-10-15(18-7-11(8-21)16(10)24
python2 /home/rfleming/work/sbgCloud/code/fork-cobratoolbox/src/analysis/thermo/componentContribution/in
Warning: getGroupVectorFromInchi did not succeed for: InChI=1/C39H51N11O18/c40-20(33(59)60)5-9-24(51)19(
python2 /home/rfleming/work/sbgCloud/code/fork-cobratoolbox/src/analysis/thermo/componentContribution/in
Warning: getGroupVectorFromInchi did not succeed for: InChI=1/C44H56N12O21/c45-21(36(66)67)5-10-26(57)43
python2 /home/rfleming/work/sbgCloud/code/fork-cobratoolbox/src/analysis/thermo/componentContribution/in
Warning: getGroupVectorFromInchi did not succeed for: InChI=1/C44H58N12O21/c45-21(36(66)67)5-10-26(57)43
python2 /home/rfleming/work/sbgCloud/code/fork-cobratoolbox/src/analysis/thermo/componentContribution/in
Warning: getGroupVectorFromInchi did not succeed for: InChI=1/C49H63N13O24/c50-22(39(71)72)5-11-28(63)34
python2 /home/rfleming/work/sbgCloud/code/fork-cobratoolbox/src/analysis/thermo/componentContribution/in
Warning: getGroupVectorFromInchi did not succeed for: InChI=1/C49H65N13O24/c50-22(39(71)72)5-11-28(63)34
```

Call the component contribution method

```
if ~isfield(model,'DfG0')
    [model,~] = componentContribution(model,training_data);
end
```

```
Running Component Contribution method
```

Setup a thermodynamically constrained model

```
if ~isfield(model,'DfGt0')
    model = setupThermoModel(model,confidenceLevel);
end
```

```
Estimating standard transformed Gibbs energies of formation.

Estimating bounds on transformed Gibbs energies.
Additional effect due to possible change in chemical potential of Hydrogen ions for transport reactions.
Additional effect due to possible change in electrical potential for transport reactions.
```

Generate a model with reactants instead of major microspecies

```
if ~isfield(model,'Srecon')
    printLevel_pHbalanceProtons=-1;
    model=pHbalanceProtons(model,massImbalance,printLevel_pHbalanceProtons,resultsBaseFileName
end
```

```
Warning: vonBertalanffy:pHbalanceProtons 'Hydrogen unbalanced reconstruction reactions
exist!
```

Determine quantitative directionality assignments

```
if ~exist('directions','var')
    fprintf('Quantitatively assigning reaction directionality.\n');
    [modelThermo, directions] = thermoConstrainFluxBounds(model,confidenceLevel,DrGt0_Uncertai
end
```

```
Quantitatively assigning reaction directionality.
9/10600 reactions with DrGtMin=DrGtMax~=0
4/10600 reactions with DrGtMin=DrGtMax=0
The following reactions have DrGtMax=DrGtMin=0:
H2Oter h2o[c]  <=> h2o[r]
H2Otn h2o[n]  <=> h2o[c]
Htr h[c]  <=> h[r]
HMR_1095 h[c]  <=> h[n]
ACYP
```

Analyse thermodynamically constrained model

Choose the cutoff for probablity that reaction is reversible

```
cumNormProbCutoff=0.2;
```

Build Boolean vectors with reaction directionality statistics

```
[modelThermo,directions]=directionalityStats(modelThermo,directions,cumNormProbCutoff,printLev
```

```
9/10600 reactions with DrGtMin=DrGtMax~=0
4/10600 reactions with DrGtMin=DrGtMax=0
Qualitative internal reaction directionality:
      8791  internal reconstruction reaction directions.
      5208  forward reconstruction assignment.
         4  reverse reconstruction assignment.
      3579  reversible reconstruction assignment.

Quantitative internal reaction directionality:
      8791  internal reconstruction reaction directions.
      7215  of which have a thermodynamic assignment.
      1572  of which have no thermodynamic assignment.
       920  forward thermodynamic only assignment.
       360  reverse thermodynamic only assignment.
      5935  reversible thermodynamic only assignment.

Qualitiative vs Quantitative:
      3065  Reversible -> Reversible
       162  Reversible -> Forward
       162  Reversible -> Reverse
       186  Reversible -> Uncertain
       758  Forward -> Forward
       198  Forward -> Reverse
      2868  Forward -> Reversible
      1384  Forward -> Uncertain
         2  Reverse -> Reverse
         0  Reverse -> Forward
         2  Reverse -> Reversible
         2  Reversible -> Uncertain

Breakdown of relaxation of reaction directionality, Qualitiative vs Quantitative:
      2868  qualitatively forward reactions that are quantitatively reversible (total).
      1419  of which are quantitatively reversible by range of dGt0. P(\Delta_{r}G^{\primeo}<0) > 0.7
        97  of which are quantitatively reversible by range of dGt0. 0.3< P(\Delta_{r}G^{\primeo}<0) < 0
      1352  of which are quantitatively reversible by range of dGt0. P(\Delta_{r}G^{\primeo}<0) < 0.3
        73  of which are quantitatively forward by fixed dGr0t, but reversible by concentration alone (z
         0  of which are quantitatively reverse by dGr0t, but reversible by concentration (negative fixe
         0  of which are quantitatively forward by dGr0t, but reversible by concentration (positve fixed
       364  of which are quantitatively reverse by dGr0t, but reversible by concentration (uncertain neg
       891  of which are quantitatively forward by dGr0t, but reversible by concentration (uncertain pos
```

```
% directions    a structue of boolean vectors with different directionality
%               assignments where some vectors contain subsets of others
%
% qualtiative -> quantiative changed reaction directions
%   .forward2Forward
%   .forward2Reverse
%   .forward2Reversible
%   .forward2Uncertain
%   .reversible2Forward
%   .reversible2Reverse
%   .reversible2Reversible
%   .reversible2Uncertain
%   .reverse2Forward
%   .reverse2Reverse
%   .reverse2Reversible
%   .reverse2Uncertain
%   .tightened
%
% subsets of qualtiatively forward  -> quantiatively reversible
%   .forward2Reversible_bydGt0
%   .forward2Reversible_bydGt0LHS
%   .forward2Reversible_bydGt0Mid
%   .forward2Reversible_bydGt0RHS
%
%   .forward2Reversible_byConc_zero_fixed_DrG0
%   .forward2Reversible_byConc_negative_fixed_DrG0
%   .forward2Reversible_byConc_positive_fixed_DrG0
%   .forward2Reversible_byConc_negative_uncertain_DrG0
%   .forward2Reversible_byConc_positive_uncertain_DrG0
```

Write out reports on directionality changes for individual reactions to the results folder.

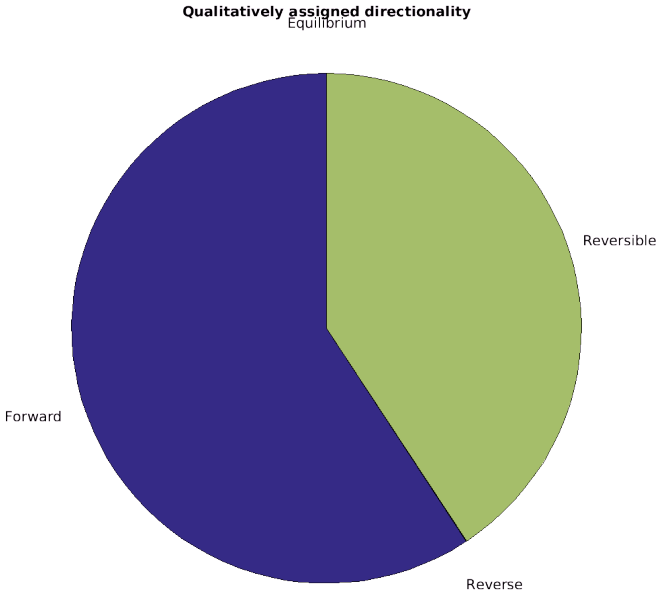
```
if 1
    fprintf('%s\n','directionalityChangeReport...');
    directionalityChangeReport(modelThermo,directions,cumNormProbCutoff,printLevel,resultsBase
end
```

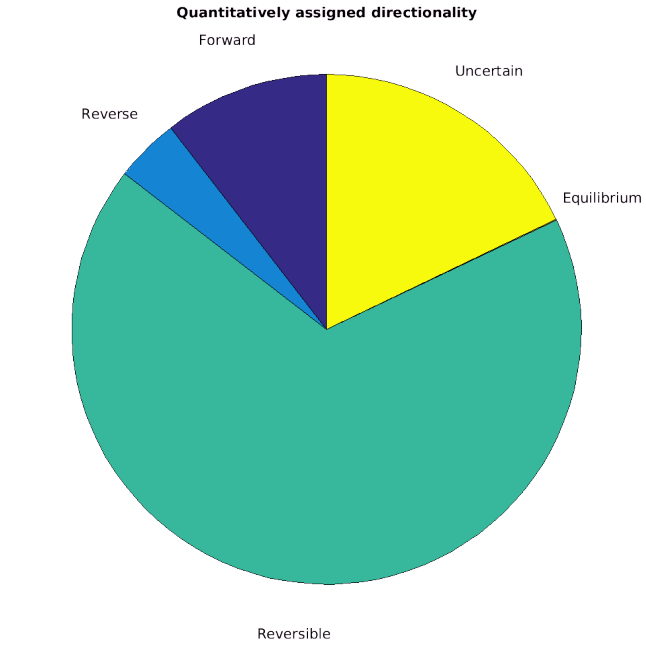
```
directionalityChangeReport...
```

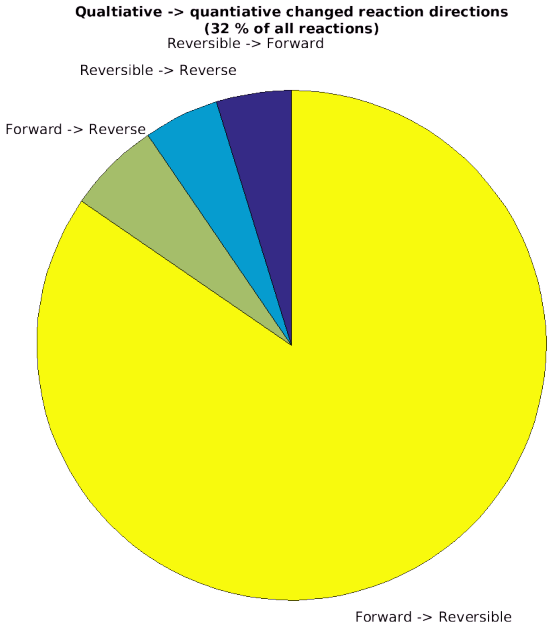
Generate pie charts with proportions of reaction directionalities and changes in directionality

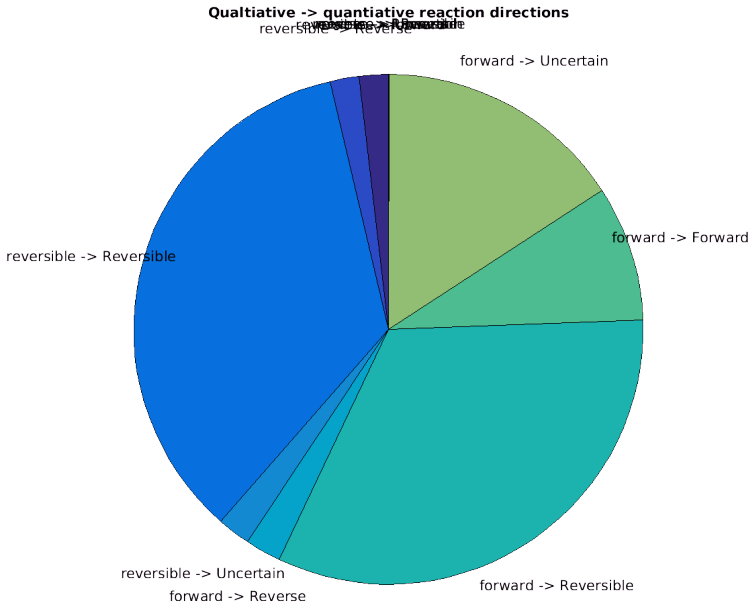
```
if 1
    fprintf('%s\n','directionalityStatFigures...');
    directionalityStatsFigures(directions,resultsBaseFileName)
end
```

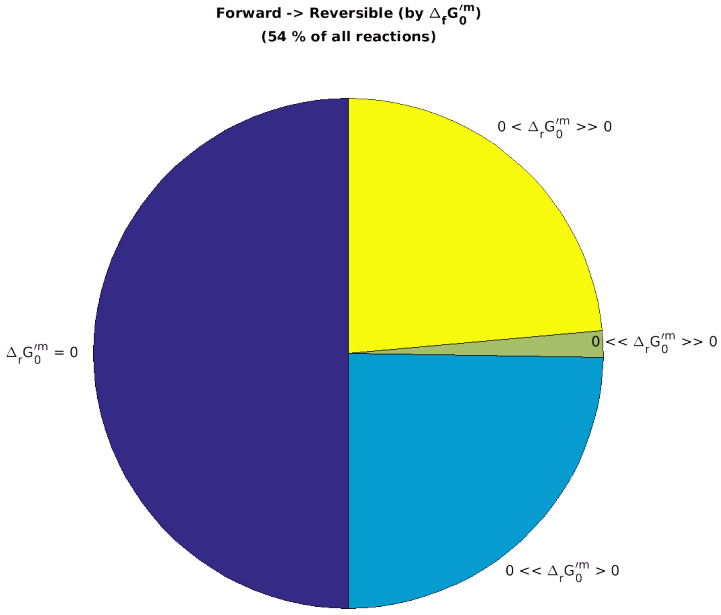
```
directionalityStatFigures...
```

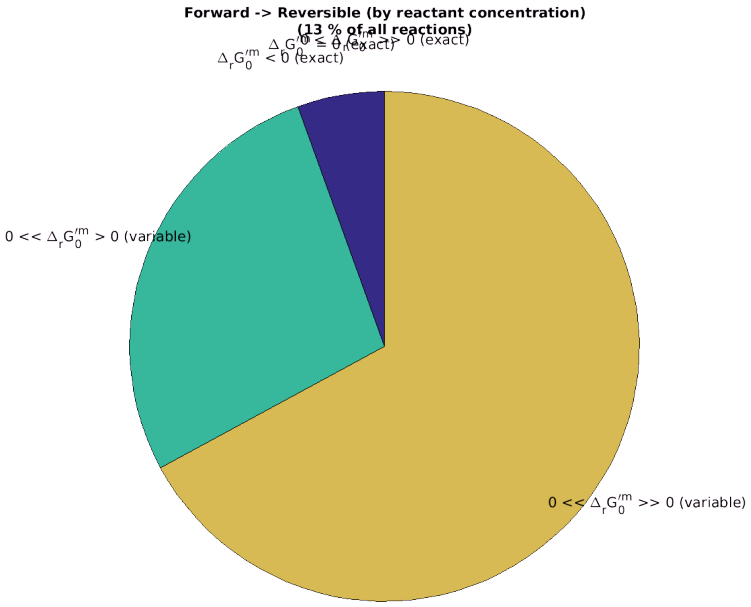

Qualitatively assigned directionality
Equilibrium
Reversible
Forward
Reverse


Quantitatively assigned directionality
Forward
Uncertain
Reverse
Equilibrium
Reversible


Qualtiative -> quantiative changed reaction directions
(32 % of all reactions)
Reversible -> Forward
Reversible -> Reverse
Forward -> Reverse
Forward -> Reversible


Qualtiative -> quantiative reaction directions
forward -> Uncertain
forward -> Forward
reversible -> Reversible
reversible -> Uncertain
forward -> Reverse
forward -> Reversible

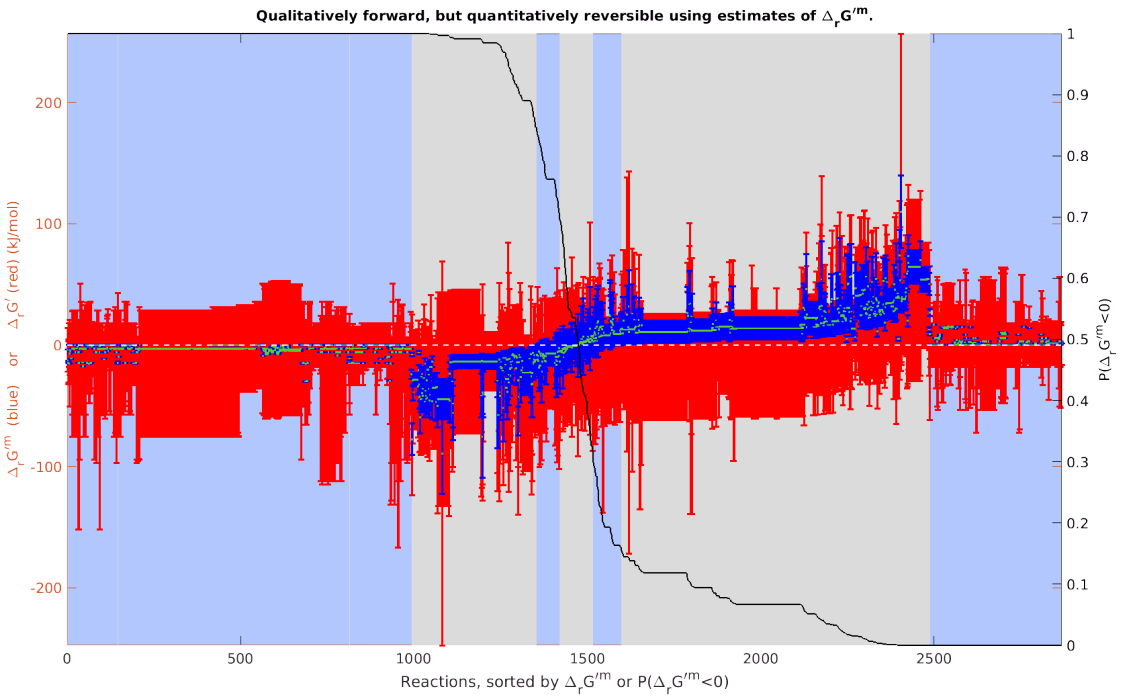




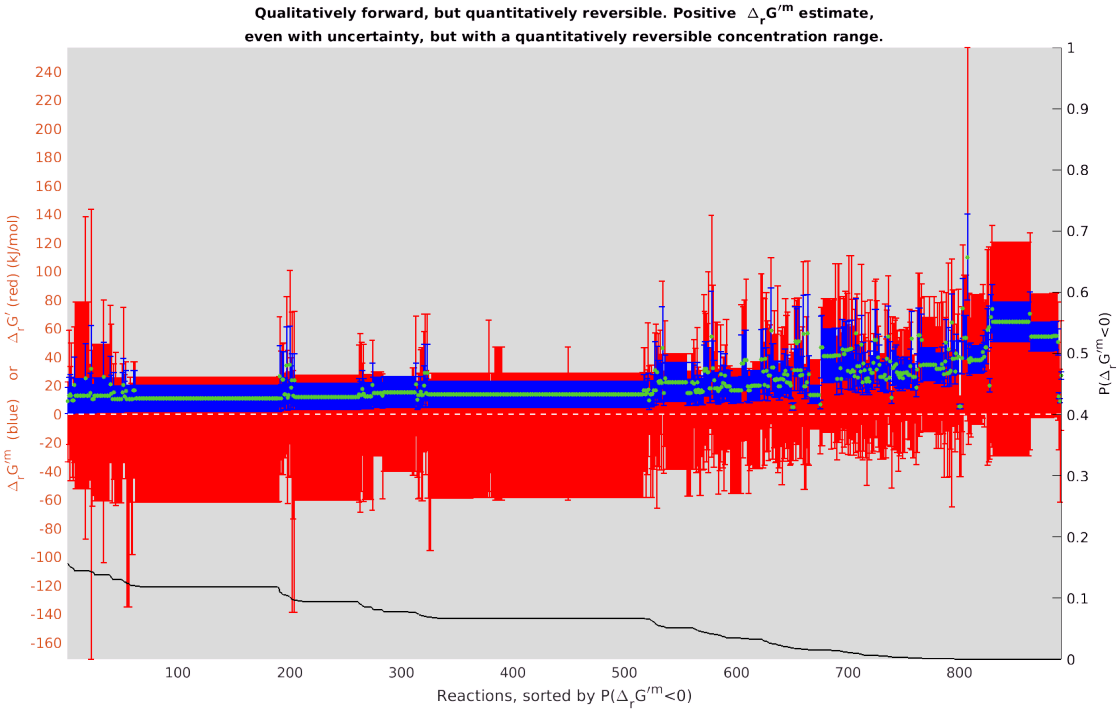
Generate figures to interpret the overall reasons for reaction directionality changes for the qualitatively forward now quantiatiavely reversible reactions

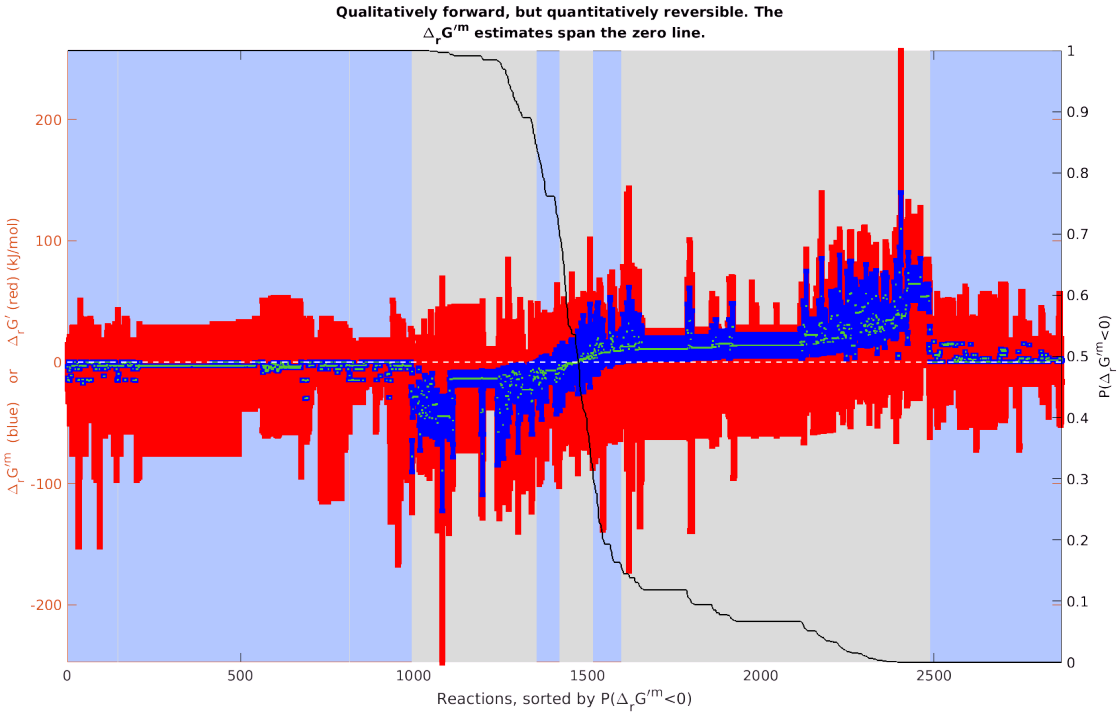
```
if any(directions.forward2Reversible)
    fprintf('%s\n','forwardReversibleFigures...');
    forwardReversibleFigures(modelThermo,directions,confidenceLevel)
end
```

```
forwardReversibleFigures...
```


Qualitatively forward, but quantitatively reversible using estimates of $\Delta_r G'^m$.
$\Delta_r G'^m$ (blue) or $\Delta_r G'$ (red) (kJ/mol)
$P(\Delta_r G'^m<0)$
Reactions, sorted by $\Delta_r G'^m$ or $P(\Delta_r G'^m<0)$


Qualitatively forward, but quantitatively reversible. Negative $\Delta_r G'^m$ estimate,
even with uncertainty, but with a quantitatively reversible concentration range.
$\Delta_r G'^m$ (blue) or $\Delta_r G'$ (red) (kJ/mol)
$P(\Delta_r G'^m<0)$
Reactions, sorted by $P(\Delta_r G'^m<0)$





Write out tables of experimental and estimated thermochemical parameters for the model

```
generateThermodynamicTables(modelThermo,resultsBaseFileName);
```

REFERENCES

[1] Fleming, R. M. T. & Thiele, I. von Bertalanffy 1.0: a COBRA toolbox extension to thermodynamically constrain metabolic models. Bioinformatics 27, 142–143 (2011).

[2] Haraldsdóttir, H. S., Thiele, I. & Fleming, R. M. T. Quantitative assignment of reaction directionality in a multicompartmental human metabolic reconstruction. Biophysical Journal 102, 1703–1711 (2012).

[3] Noor, E., Haraldsdóttir, H. S., Milo, R. & Fleming, R. M. T. Consistent Estimation of Gibbs Energy Using Component Contributions. PLoS Comput Biol 9, e1003098 (2013).

[4] Fleming, R. M. T. , Predicat, G., Haraldsdóttir, H. S., Thiele, I. von Bertalanffy 2.0 (in preparation).