Modular Modelling of the *e.coli* core model

Peter Gawthrop. *peter.gawthrop@unimelb.edu.au*

January 15, 2021

Contents

1 Introduction

The Network Thermodynamics/Bond Graph approach of Oster et al. (1971, 1973) extended by Gawthrop and Crampin (2016, 2014, 2017) to modelling biomolecular systems of interest to systems biologists developed independently from the stoichiometric approach Palsson (2006, 2011, 2015).

However, the conceptual point of intersection of the two approaches is the fact that the stoichiometric matrix is the modulus of the conceptual multiport transformer linking reactions

to species. This was pointed out by Cellier and Greifeneder (2009). This means that the two approaches are complementary and each can build on the strengths of the other.

This notebook focuses on building modular models of metabolism and consequent pathway analysis based on the Escherichia coli Core Model Orth et al. (2010a); energetic issues are not explicitly considered here although they are implicit in the bond graph models.

1.1 Strengths of the stoichiometric approach

1. The stoichiometric matrix provides a precise description of the structure of chemical reaction networks.
2. Stoichiometric matrices are available for biologically significant systems including whole-cell models.
3. Powerful linear algebra concepts such as matrix null spaces and singular-value decompositions can be applied.
4. It provides the basis for flux-balance analysis Orth et al. (2010b).

1.2 Strengths of the bond graph approach

1. Bond graph provide an energy-based approach to modelling.
2. The bond graph model can be analysed for energy flows and efficiency as described by Gawthrop and Crampin (2018).
3. The bond graph approach is multi domain and can thus, for example, model electrochemical systems including neurodynamics Gawthrop et al. (2017), redox reactions Gawthrop (2017a) and transporters Pan et al. (2019).
4. The bond graph approch is modular: subsystems can be connected using energy ports Gawthrop and Crampin (2016); Gawthrop et al. (2015).
5. The chemostat concept Gawthrop and Crampin (2016) gives a convenient and flexible way of turning a closed system into an open system and analysing the concomitant pathway structure.
6. BondGraphTools provide a symbolic basis for describing and analysing bondgraphs within Python.

1.3 Import some python code

The bond graph analysis uses a number of Python modules:

[1]:
```
## Paths
NeedPath=True
if NeedPath:
    import sys
    sys.path += ['/usr/lib/python3/dist-packages']
```

[2]:
```
import BondGraphTools as bgt
import numpy as np
import IPython.display as disp

## Stoichiometric analysis
import stoich as st

## Export stoichiometry as bond graph
import stoichBondGraph as stbg
```

```
## Modular bond graphs
import modularBondGraph as mbg

## Extract stoichiometry from a CobraPy model
import CobraExtract as Extract

## Control outputs
quiet = True
chemformula = True
```

[3]:
```
def printChem(chem):
    Chem = ''
    for c in chem:
        Chem += f'\ch{{{c}}}, '

    return Chem[:-2]
```

1.4 Combining the two approaches

The key stoichiometric concept of pathways has already been given a bond graph interpretation Gawthrop (2017b); Gawthrop and Crampin (2017)

This note shows how:

1. bond graphs can be created from stoichiometric data using the ecoli core model Orth et al. (2010a) as an example.
2. bond graphs of subsystems can be extracted from such stoichiometric data
3. pathways can be anaysed and behavior elucidated using appropriate chice of chemostats
4. bond graphs of the extracted subsystems can be recombined in a modular fashion

Much remains to be done in exploiting the combination of the two approaches including:

1. A new look at flux-balance analysis.
2. Further work on energy and efficiency of cellular systems.
3. Using the large stoichiometric models available either directly or extracted from models in SBML or cellML format.

1.5 External metabolites

The standard stoichiometric approach is to create open systems from closed systems by adding "dangling reactions" to species which connect to the outside world as external metabolites– for example: ATP $\rightleftharpoons$. In contrast, the bond graph approacj would declare ATP to be a chemostat. Thus when extracting a bond graph from a stoichiometric model, dangling reactions are deleted and the corresponding species added to a list of chemostats.

Chemostats provide a more flexible approach as they can be created without changing system structure.

In the following discussion, chemostats are added as appropriate to illustrate the various pathways.

2 Extract various modules from the ecoli core model

2.1 Extract full ecoli core model

In this case the ecoli core model is extracted from the CobraPy model: 'textbook'. This model corresponds to that discussed in the the textbook Palsson (2015). An integer version of the stoichiometric matrix, together with reactions and species is produced. Note that the reaction labeled 'Biomass_Ecoli_core' is not a reaction but is associated with the optimisation procedure - it can be ignored in this notebook.

Names of reactions and species are converted to upper case and the *c* (cytosol) subscript removed for clarity.

[4]:
```
sm = Extract.extract(Remove=['_C','__' ],
                     ␣
 ↪negReaction=['RPI','PGK','PGM','SUCOAS','FRD7'],quiet=quiet)
```

```
Extracting stoichiometric matrix from: textbook
Cobra Model name: e_coli_core BondGraphTools name: e_coli_core_abg
Extract.Integer only handles one non-integer per reaction
Multiplying reaction BIOMASS_ECOLIORE ( 12 ) by 0.6684491978609626 to avoid non-
integer species 3PG ( 2 )
Multiplying reaction CYTBD ( 15 ) by 2.0 to avoid non-integer species O2 ( 55 )
Multiplying reaction FRD7 ( 23 ) by -1
Multiplying reaction PGK ( 54 ) by -1
Multiplying reaction PGM ( 56 ) by -1
Multiplying reaction RPI ( 65 ) by -1
Multiplying reaction SUCOAS ( 69 ) by -1
```

2.2 Extract Glycolysis

[5]:
```
reaction =␣
 ↪['GLCPTS','PGI','PFK','FBP','FBA','TPI','GAPD','PGK','PGM','ENO','PYK']
s0 = Extract.choose(sm,reaction=reaction)
print('Reactions:', reaction)
disp.Latex(st.sprintrl(s0,chemformula=True))
```

```
Reactions: ['GLCPTS', 'PGI', 'PFK', 'FBP', 'FBA', 'TPI', 'GAPD', 'PGK', 'PGM',
'ENO', 'PYK']
```

[5]:

$$\mathrm{GLCD_E} + \mathrm{PEP} \xrightleftharpoons{\mathrm{GLCPTS}} \mathrm{G_6P} + \mathrm{PYR} \tag{1}$$

$$\mathrm{G_6P} \xrightleftharpoons{\mathrm{PGI}} \mathrm{F_6P} \tag{2}$$

$$\mathrm{ATP} + \mathrm{F_6P} \xrightleftharpoons{\mathrm{PFK}} \mathrm{ADP} + \mathrm{FDP} + \mathrm{H} \tag{3}$$

$$\mathrm{FDP} + \mathrm{H_2O} \xrightleftharpoons{\mathrm{FBP}} \mathrm{F_6P} + \mathrm{PI} \tag{4}$$

$$\mathrm{FDP} \xrightleftharpoons{\mathrm{FBA}} \mathrm{DHAP} + \mathrm{G_3P} \tag{5}$$

$$\mathrm{DHAP} \xrightleftharpoons{\mathrm{TPI}} \mathrm{G_3P} \tag{6}$$

$$\mathrm{G_3P} + \mathrm{NAD} + \mathrm{PI} \xrightleftharpoons{\mathrm{GAPD}} {}_{13}\mathrm{DPG} + \mathrm{H} + \mathrm{NADH} \tag{7}$$

$${}_{13}\mathrm{DPG} + \mathrm{ADP} \xrightleftharpoons{\mathrm{PGK}} {}_{3}\mathrm{PG} + \mathrm{ATP} \tag{8}$$

$${}_{3}\mathrm{PG} \xrightleftharpoons{\mathrm{PGM}} {}_{2}\mathrm{PG} \tag{9}$$

$${}_{2}\mathrm{PG} \xrightleftharpoons{\mathrm{ENO}} \mathrm{H_2O} + \mathrm{PEP} \tag{10}$$

$$\mathrm{ADP} + \mathrm{H} + \mathrm{PEP} \xrightleftharpoons{\mathrm{PYK}} \mathrm{ATP} + \mathrm{PYR} \tag{11}$$

[6]:
```
s0['name'] = 'GLY_abg'
stbg.model(s0)
import GLY_abg
s = st.stoich(GLY_abg.model(),quiet=True)
## Sanity check
err = np.linalg.norm(s['N']-s0['N'])
print("Error:",err)
```

```
Error: 0.0
```

2.2.1 Pathway analysis

[7]:
```
chemostats0 = ['ADP','ATP','H2O','PI','H']
sc0 = st.statify(s,chemostats=chemostats0)
chemostats = ['GLCD_E','PYR','NAD','NADH']
chemostats.extend(chemostats0)
sc = st.statify(s,chemostats=chemostats)
print(st.sprintp(sc))
sp = st.path(s,sc)
disp.Latex(st.sprintrl(sp,chemformula=True))
```

```
2 pathways
0:  + PFK + FBP
1:  + GLCPTS + PGI + PFK + FBA + TPI + 2 GAPD + 2 PGK + 2 PGM + 2 ENO + PYK
```

[7]:

$$\mathrm{ATP} + \mathrm{H_2O} \xrightleftharpoons{\mathrm{pr_1}} \mathrm{ADP} + \mathrm{H} + \mathrm{PI} \tag{12}$$

$$2\,\mathrm{ADP} + \mathrm{GLCD_E} + 2\,\mathrm{NAD} + 2\,\mathrm{PI} \xrightleftharpoons{\mathrm{pr_2}} 2\,\mathrm{ATP} + 2\,\mathrm{H} + 2\,\mathrm{H_2O} + 2\,\mathrm{NADH} + 2\,\mathrm{PYR} \tag{13}$$

Pathway reactions:

- A futile cycle consuming ATP
- A pathway from $GLCD_E$ to PYR

2.3 Extract TCA part of full model

A list of the reactions of the TCA cycle is provided and the corresponding species and stoichiometric matrix are extracted. Note that the NAD/NADP interconversion reaction NADTRHD is included.

[8]:
```
reaction = ['CS','ACONTA','ACONTB','ICDHYR','AKGDH','SUCOAS',
                'FRD7',
                 'SUCDI','FUM','MDH',
        'NADTRHD']
s0 = Extract.choose(sm,reaction=reaction)
disp.Latex(st.sprintrl(s0,chemformula=True))
```

[8]:

$$\mathrm{ACCOA + H_2O + OAA \xrightleftharpoons{CS} CIT + COA + H} \tag{14}$$
$$\mathrm{CIT \xrightleftharpoons{ACONTA} ACON + H_2O} \tag{15}$$
$$\mathrm{ACON + H_2O \xrightleftharpoons{ACONTB} ICIT} \tag{16}$$
$$\mathrm{ICIT + NADP \xrightleftharpoons{ICDHYR} AKG + CO_2 + NADPH} \tag{17}$$
$$\mathrm{AKG + COA + NAD \xrightleftharpoons{AKGDH} CO_2 + NADH + SUCCOA} \tag{18}$$
$$\mathrm{ADP + PI + SUCCOA \xrightleftharpoons{SUCOAS} ATP + COA + SUCC} \tag{19}$$
$$\mathrm{Q_8 + SUCC \xrightleftharpoons{FRD_7} FUM + Q_8H_2} \tag{20}$$
$$\mathrm{Q_8 + SUCC \xrightleftharpoons{SUCDI} FUM + Q_8H_2} \tag{21}$$
$$\mathrm{FUM + H_2O \xrightleftharpoons{FUM} MALL} \tag{22}$$
$$\mathrm{MALL + NAD \xrightleftharpoons{MDH} H + NADH + OAA} \tag{23}$$

$$\mathrm{NAD + NADPH \xrightleftharpoons{NADTRHD} NADH + NADP} \tag{24}$$

2.3.1 Create bond graph and recreate stoichiometry

The bond graph TCA_abg is created and written to TCA_abg.py from whence it can me imported. The stoichiometric matrix generated from TCA_abg.model() is compared with that extracted from the ecoli core model to check that all is working correctely.

[9]:
```
s0['name'] = 'TCA_abg'
stbg.model(s0)
import TCA_abg
s = st.stoich(TCA_abg.model(),quiet=True)

## Sanity check
err = np.linalg.norm(s['N']-s0['N'])
print("Error:",err)
```

```
Error: 0.0
```

2.3.2 Pathway analysis

Thee species corresponding to ATP hydrolysis, NAD/NADH, NADP/NADPH and Q8/Q8H2 (ubiquone, but maybe FAD, I think) are set as the basic chemostats in the list chemostats0.

The overall reaction of the TCA cycle converts ACCOA to COA and CO_2 Garrett and Grisham (2017), so these three species are also set as chemostats.

[10]:
```
print('No chemostats')
print(st.sprintp(s))
chemostats0 = ['ADP', 'ATP', 'H2O', 'NAD', 'NADH', 'PI','H'
               ,'Q8','Q8H2']
sc0 = st.statify(s,chemostats=chemostats0)
print('Basic chemostats',chemostats0)
print(st.sprintp(sc0))
chemostats = ['ACCOA','COA','CO2']
chemostats.extend(chemostats0)
sc = st.statify(s,chemostats=chemostats)
print('Chemostats',chemostats)
print(st.sprintp(sc))
```

```
No chemostats
1 pathways
0:  - FRD7 + SUCDI

Basic chemostats ['ADP', 'ATP', 'H2O', 'NAD', 'NADH', 'PI', 'H', 'Q8', 'Q8H2']
1 pathways
0:  - FRD7 + SUCDI

Chemostats ['ACCOA', 'COA', 'CO2', 'ADP', 'ATP', 'H2O', 'NAD', 'NADH', 'PI',
'H', 'Q8', 'Q8H2']
2 pathways
0:  - FRD7 + SUCDI
1:  + CS + ACONTA + ACONTB + ICDHYR + AKGDH + SUCOAS + FRD7 + FUM + MDH +
NADTRHD
```

With only the basic chemostats, there is one internal cycle. This performs no conversions and so the reaction is empty.

[11]:
```
sp0 = st.path(s,sc0)
disp.Latex(st.sprintrl(sp0,chemformula=True,split=10))
```

[11]:

$$\xrightleftharpoons{\mathrm{pr}_1} \tag{25}$$

When the chemostats ['ACCOA','COA','CO2'] are added, the entire TCA cycle appears as a pathway and the overall reaction is

[12]:
```
sp = st.path(s,sc)
disp.Latex(st.sprintrl(sp,chemformula=True,split=10))
```

[12]:

$$\xrightleftharpoons{pr_1} \tag{26}$$

$$\mathrm{ACCOA + ADP + 2\,H_2O + 3\,NAD + PI + Q_8} \xrightleftharpoons{pr_2} \mathrm{ATP + 2\,CO_2 + COA + 2\,H + 3\,NADH + Q_8H_2} \tag{27}$$

Pathway reactions:

- A null cycle (shared with closed system)
- Pathway from ACCOA to CO2

2.4 Include Pyruvate (Pyr) reactions: PDH and PFL

To take this example further, include the two reactions converting pyruvate to ACCOA.

[13]:
```
reaction.extend(['PDH','PFL'])
s0 = Extract.choose(sm,reaction=reaction)
disp.Latex(st.sprintrl(s0,chemformula=True))
```

[13]:

$$\mathrm{ACCOA + H_2O + OAA} \xrightleftharpoons{CS} \mathrm{CIT + COA + H} \tag{28}$$

$$\mathrm{CIT} \xrightleftharpoons{ACONTA} \mathrm{ACON + H_2O} \tag{29}$$

$$\mathrm{ACON + H_2O} \xrightleftharpoons{ACONTB} \mathrm{ICIT} \tag{30}$$

$$\mathrm{ICIT + NADP} \xrightleftharpoons{ICDHYR} \mathrm{AKG + CO_2 + NADPH} \tag{31}$$

$$\mathrm{AKG + COA + NAD} \xrightleftharpoons{AKGDH} \mathrm{CO_2 + NADH + SUCCOA} \tag{32}$$

$$\mathrm{ADP + PI + SUCCOA} \xrightleftharpoons{SUCOAS} \mathrm{ATP + COA + SUCC} \tag{33}$$

$$\mathrm{Q_8 + SUCC} \xrightleftharpoons{FRD_7} \mathrm{FUM + Q_8H_2} \tag{34}$$

$$\mathrm{Q_8 + SUCC} \xrightleftharpoons{SUCDI} \mathrm{FUM + Q_8H_2} \tag{35}$$

$$\mathrm{FUM + H_2O} \xrightleftharpoons{FUM} \mathrm{MALL} \tag{36}$$

$$\mathrm{MALL + NAD} \xrightleftharpoons{MDH} \mathrm{H + NADH + OAA} \tag{37}$$

$$\mathrm{NAD + NADPH} \xrightleftharpoons{NADTRHD} \mathrm{NADH + NADP} \tag{38}$$

$$\mathrm{COA + NAD + PYR} \xrightleftharpoons{PDH} \mathrm{ACCOA + CO_2 + NADH} \tag{39}$$

$$\mathrm{COA + PYR} \xrightleftharpoons{PFL} \mathrm{ACCOA + FOR} \tag{40}$$

2.4.1 Create bond graph and recreate stoichiometry

[14]:
```
s0['name'] = 'PyrTCA_abg'
stbg.model(s0)
import PyrTCA_abg
```

```
s = st.stoich(PyrTCA_abg.model(),quiet=True)

## Sanity check
err = np.linalg.norm(s['N']-s0['N'])
print("Error:",err)
```

```
Error: 0.0
```

2.4.2 Pathway analysis

The relevant chemostats are now the substrate PYR and the product CO_2 together with the reactions ATP hydrolysis, NAD/NADH, NADP/NADPH and Q8/Q8H2 andthe additional product FOR

[15]:
```
print('No chemostats')
print(st.sprintp(s,removeSingle=True))
chemostats0 = ['ADP', 'ATP', 'H2O', 'NAD', 'NADH', 'PI','NADP', 'NADPH','H'
               ,'Q8','Q8H2','FOR']
sc0 = st.statify(s,chemostats=chemostats0)
print('Basic chemostats',chemostats0)
print(st.sprintp(sc0,removeSingle=True))
chemostats = ['PYR','CO2']
chemostats.extend(chemostats0)
sc = st.statify(s,chemostats=chemostats)
print('Chemostats:', printChem(chemostats))
print(st.sprintp(sc,removeSingle=True))
```

```
No chemostats
1 non-unit pathways
0:  - FRD7 + SUCDI

Basic chemostats ['ADP', 'ATP', 'H2O', 'NAD', 'NADH', 'PI', 'NADP', 'NADPH',
'H', 'Q8', 'Q8H2', 'FOR']
1 non-unit pathways
0:  - FRD7 + SUCDI

Chemostats: \ch{PYR}, \ch{CO2}, \ch{ADP}, \ch{ATP}, \ch{H2O}, \ch{NAD},
\ch{NADH}, \ch{PI}, \ch{NADP}, \ch{NADPH}, \ch{H}, \ch{Q8}, \ch{Q8H2}, \ch{FOR}
3 non-unit pathways
0:  - FRD7 + SUCDI
1:  + CS + ACONTA + ACONTB + ICDHYR + AKGDH + SUCOAS + FRD7 + FUM + MDH + PDH
2:  + CS + ACONTA + ACONTB + ICDHYR + AKGDH + SUCOAS + FRD7 + FUM + MDH + PFL
```

[16]:
```
sp = st.path(s,sc)
disp.Latex(st.sprintrl(sp,chemformula=True,split=10))
```

[16]:

$$\xrightleftharpoons{pr_1} \tag{41}$$

$$ADP + 2\,H_2O + 3\,NAD + NADP + PI + PYR + Q_8 \xrightleftharpoons{pr_2} ATP + 3\,CO_2 + 2\,H + 3\,NADH + NADPH + Q_8H_2 \tag{42}$$

$$ADP + 2\,H_2O + 2\,NAD + NADP + PI + PYR + Q_8 \xrightleftharpoons{pr_3} ATP + 2\,CO_2 + FOR + 2\,H + 2\,NADH + NADPH \tag{43}$$

Pathway reactions:

- A null cycle (shared with closed system)
- Pathway from PYR to CO2 via reaction PDH
- Pathway from PYR to CO2 via reaction PFL

2.5 Extract the Electron Transport Chain (ETC)

[17]:
```
reaction = ['NADH16','CYTBD']
s0 = Extract.choose(sm,reaction=reaction)
disp.Latex(st.sprintrl(s0,chemformula=True))
```

[17]:

$$4\,H + NADH + Q_8 \xrightleftharpoons{NADH_{16}} 3\,H_E + NAD + Q_8H_2 \tag{44}$$

$$4\,H + O_2 + 2\,Q_8H_2 \xrightleftharpoons{CYTBD} 2\,H_2O + 4\,H_E + 2\,Q_8 \tag{45}$$

[18]:
```
s0['name'] = 'ETC_abg'
stbg.model(s0)
import ETC_abg
##imp.reload(ETC_abg)
s = st.stoich(ETC_abg.model(),quiet=True)
err = np.linalg.norm(s['N']-s0['N'])
print("Error:",err)
s['reaction'] = s0['reaction']
```

```
Error: 0.0
```

2.5.1 Pathway analysis

[19]:
```
print('No chemostats')
print(st.sprintp(s))
chemostats = ['NADH','NAD','O2','H2O','H','H_E']
sc = st.statify(s,chemostats=chemostats)
print('Chemostats',chemostats)
print(st.sprintp(sc))
```

```
No chemostats
0 pathways

Chemostats ['NADH', 'NAD', 'O2', 'H2O', 'H', 'H_E']
1 pathways
```

```
0:  + 2 NADH16 + CYTBD
```

[20]:
```
sp = st.path(s,sc)
disp.Latex(st.sprintrl(sp,chemformula=True))
```

[20]:

$$12\,\mathrm{H} + 2\,\mathrm{NADH} + \mathrm{O_2} \overset{\mathrm{pr_1}}{\rightleftharpoons} 2\,\mathrm{H_2O} + 10\,\mathrm{H_E} + 2\,\mathrm{NAD} \tag{46}$$

Pathway reactions:

- Pathway from NADH and O2 to NAD and H2O pumping 10 protons

2.6 Extract ATPase

[21]:
```
reaction = ['ATPS4R']
s0 = Extract.choose(sm,reaction=reaction)
disp.Latex(st.sprintrl(s0,chemformula=True))
```

[21]:

$$\mathrm{ADP} + 4\,\mathrm{H_E} + \mathrm{PI} \overset{\mathrm{ATPS_4R}}{\rightleftharpoons} \mathrm{ATP} + 3\,\mathrm{H} + \mathrm{H_2O} \tag{47}$$

[22]:
```
s0['name'] = 'ATP_abg'
stbg.model(s0)
import ATP_abg
##imp.reload(ATP_abg)
s = st.stoich(ATP_abg.model(),quiet=True)
err = np.linalg.norm(s['N']-s0['N'])
print("Error:",err)
s['reaction'] = s0['reaction']
```

```
Error: 0.0
```

2.6.1 Pathway analysis

[23]:
```
print('No chemostats')
print(st.sprintp(s))
chemostats = ['H','H_E','ATP','ADP','PI','H2O']
sc = st.statify(s,chemostats=chemostats)
print('Chemostats',chemostats)
print(st.sprintp(sc,removeSingle=False))
```

```
No chemostats
0 pathways

Chemostats ['H', 'H_E', 'ATP', 'ADP', 'PI', 'H2O']
1 pathways
0:  + ATPS4R
```

[24]:
```
sp = st.path(s,sc,removeSingle=False)
disp.Latex(st.sprintrl(sp,chemformula=True))
```

[24]:

$$\mathrm{ADP} + 4\,\mathrm{H_E} + \mathrm{PI} \xrightleftharpoons{\mathrm{pr_1}} \mathrm{ATP} + 3\,\mathrm{H} + \mathrm{H_2O} \tag{48}$$

Pathway reactions:

- Pathway from ADP to ATP pumped by 3 protons.

3 Modularity: the Glycolysis/TCA network

To examine the glycolysis/TCA network, it is convenient and informative to take a modular approach: the glycolysis network and the TCA network are extracted as separate subsystems and then combined. The example shows two approaches to combining the subsystems:

1. combining the stoichiometric matrices
2. combining the bond graphs.

3.1 Create composite model

[25]:
```
GlyTCA = bgt.new(name='GlyTCA')    # Create system
Gly = GLY_abg.model()
TCA = PyrTCA_abg.model()
GlyTCA.add(Gly,TCA)
```

3.1.1 Unify common species

PYR is produced by Gly and consumed by TCA. ATP, ADP, PI, H, NAD, NADH , H2O are common to both modules.

[26]:
```
common = ['PYR','ATP','ADP','PI','H','NAD','NADH','H2O']
print('Common species:', '{'+printChem(common)+'}')
mbg.unify(GlyTCA,common=common,quiet=quiet)
```

```
Common species: {\ch{PYR}, \ch{ATP}, \ch{ADP}, \ch{PI}, \ch{H}, \ch{NAD},
\ch{NADH}, \ch{H2O}}
```

[27]:
```
## Stoichiometry of unified module
s = st.stoich(GlyTCA,quiet=True)
```

[28]:
```
## Create BG
s['name'] = 'GlyTCA_abg'
stbg.model(s)
import GlyTCA_abg
```

3.1.2 Pathway analysis

[29]:
```
chemostats0 = ['ADP', 'ATP', 'H2O', 'NAD', 'NADH', 'PI','H'
               ,'Q8','Q8H2','FOR']
chemostats = ['GLCD_E','CO2']
chemostats.extend(chemostats0)
sc = st.statify(s,chemostats=chemostats)
print('Chemostats:', printChem(chemostats))
print(st.sprintp(sc))
```

```
Chemostats: \ch{GLCD_E}, \ch{CO2}, \ch{ADP}, \ch{ATP}, \ch{H2O}, \ch{NAD},
\ch{NADH}, \ch{PI}, \ch{H}, \ch{Q8}, \ch{Q8H2}, \ch{FOR}
4 pathways
0:  + PFK + FBP
1:  - FRD7 + SUCDI
2:  + GLCPTS + PGI + PFK + FBA + TPI + 2 GAPD + 2 PGK + 2 PGM + 2 ENO + PYK + 2
CS + 2 ACONTA + 2 ACONTB + 2 ICDHYR + 2 AKGDH + 2 SUCOAS + 2 FRD7 + 2 FUM + 2
MDH + 2 NADTRHD + 2 PDH
3:  + GLCPTS + PGI + PFK + FBA + TPI + 2 GAPD + 2 PGK + 2 PGM + 2 ENO + PYK + 2
CS + 2 ACONTA + 2 ACONTB + 2 ICDHYR + 2 AKGDH + 2 SUCOAS + 2 FRD7 + 2 FUM + 2
MDH + 2 NADTRHD + 2 PFL
```

[30]:
```
sp = st.path(s,sc)
disp.Latex(st.sprintrl(sp,chemformula=True))
```

[30]:

$$\mathrm{ATP} + \mathrm{H_2O} \xrightleftharpoons{\mathrm{pr_1}} \mathrm{ADP} + \mathrm{PI} + \mathrm{H} \tag{49}$$

$$\xrightleftharpoons{\mathrm{pr_2}} \tag{50}$$

$$\mathrm{GLCD_E} + 2\,\mathrm{Q_8} + 4\,\mathrm{ADP} + 4\,\mathrm{PI} + 10\,\mathrm{NAD} + 2\,\mathrm{H_2O} \xrightleftharpoons{\mathrm{pr_3}} 6\,\mathrm{CO_2} + 2\,\mathrm{Q_8H_2} + 4\,\mathrm{ATP} + 6\,\mathrm{H} + 10\,\mathrm{NADH} \tag{51}$$

$$\mathrm{GLCD_E} + 2\,\mathrm{Q_8} + 4\,\mathrm{ADP} + 4\,\mathrm{PI} + 8\,\mathrm{NAD} + 2\,\mathrm{H_2O} \xrightleftharpoons{\mathrm{pr_4}} 4\,\mathrm{CO_2} + 2\,\mathrm{FOR} + 2\,\mathrm{Q_8H_2} + 4\,\mathrm{ATP} + 6\,\mathrm{H} + 8\,\mathrm{NADH} \tag{52}$$

Pathway reactions:

- A futile cycle
- A null cycle
- Pathway from GLCD_E to CO2 via reaction PDH
- Pathway from GLCD_E to CO2 via reaction PFL

4 Modularity: metabolism

The above methods are brought together to generate the metabolic pathways by combining the modules describing: Glycolysis, the TCA cycle, the Electron Transport Chain and ATPase. Once again, the overall bond graph is created by combining modules and unifying common species.

[31]:
```
Met = bgt.new(name='Met')    # Create systemGlyTCA
GlyTCA = GlyTCA_abg.model()
ETC = ETC_abg.model()
ATP = ATP_abg.model()
Met.add(GlyTCA,ETC,ATP)
```

[32]:
```
## Unify species common to modules
common = ['ATP','ADP','PI','H','H_E','NAD','NADH','H2O','Q8','Q8H2']
print('Common species:', '{'+printChem(common)+'}')
mbg.unify(Met,common=common,quiet=True)
```

```
Common species: {\ch{ATP}, \ch{ADP}, \ch{PI}, \ch{H}, \ch{H_E}, \ch{NAD},
\ch{NADH}, \ch{H2O}, \ch{Q8}, \ch{Q8H2}}
```

4.1 Pathway analysis

[33]:
```
s = st.stoich(Met,quiet=True)
disp.Latex(st.sprintrl(s,chemformula=True))
```

[33]:

$$\mathrm{GLCD_E} + \mathrm{PEP} \xrightleftharpoons{\mathrm{GLCPTS}} \mathrm{G_6P} + \mathrm{PYR} \tag{53}$$

$$\mathrm{G_6P} \xrightleftharpoons{\mathrm{PGI}} \mathrm{F_6P} \tag{54}$$

$$\mathrm{F_6P} + \mathrm{ATP} \xrightleftharpoons{\mathrm{PFK}} \mathrm{FDP} + \mathrm{ADP} + \mathrm{H} \tag{55}$$

$$\mathrm{FDP} + \mathrm{H_2O} \xrightleftharpoons{\mathrm{FBP}} \mathrm{F_6P} + \mathrm{PI} \tag{56}$$

$$\mathrm{FDP} \xrightleftharpoons{\mathrm{FBA}} \mathrm{DHAP} + \mathrm{G_3P} \tag{57}$$

$$\mathrm{DHAP} \xrightleftharpoons{\mathrm{TPI}} \mathrm{G_3P} \tag{58}$$

$$\mathrm{G_3P} + \mathrm{PI} + \mathrm{NAD} \xrightleftharpoons{\mathrm{GAPD}} {}_{13}\mathrm{DPG} + \mathrm{H} + \mathrm{NADH} \tag{59}$$

$${}_{13}\mathrm{DPG} + \mathrm{ADP} \xrightleftharpoons{\mathrm{PGK}} {}_{3}\mathrm{PG} + \mathrm{ATP} \tag{60}$$

$${}_{3}\mathrm{PG} \xrightleftharpoons{\mathrm{PGM}} {}_{2}\mathrm{PG} \tag{61}$$

$${}_{2}\mathrm{PG} \xrightleftharpoons{\mathrm{ENO}} \mathrm{PEP} + \mathrm{H_2O} \tag{62}$$

$$\mathrm{PEP} + \mathrm{ADP} + \mathrm{H} \xrightleftharpoons{\mathrm{PYK}} \mathrm{PYR} + \mathrm{ATP} \tag{63}$$

$$\mathrm{ACCOA} + \mathrm{OAA} + \mathrm{H_2O} \xrightleftharpoons{\mathrm{CS}} \mathrm{CIT} + \mathrm{COA} + \mathrm{H} \tag{64}$$

$$\mathrm{CIT} \xrightleftharpoons{\mathrm{ACONTA}} \mathrm{ACON} + \mathrm{H_2O} \tag{65}$$

$$\mathrm{ACON} + \mathrm{H_2O} \xrightleftharpoons{\mathrm{ACONTB}} \mathrm{ICIT} \tag{66}$$

$$\mathrm{ICIT} + \mathrm{NADP} \xrightleftharpoons{\mathrm{ICDHYR}} \mathrm{AKG} + \mathrm{CO_2} + \mathrm{NADPH} \tag{67}$$

$$\mathrm{AKG} + \mathrm{COA} + \mathrm{NAD} \xrightleftharpoons{\mathrm{AKGDH}} \mathrm{CO_2} + \mathrm{SUCCOA} + \mathrm{NADH} \tag{68}$$

$$\mathrm{SUCCOA} + \mathrm{ADP} + \mathrm{PI} \xrightleftharpoons{\mathrm{SUCOAS}} \mathrm{COA} + \mathrm{SUCC} + \mathrm{ATP} \tag{69}$$

$$\mathrm{SUCC} + \mathrm{Q_8} \xrightleftharpoons{\mathrm{FRD_7}} \mathrm{FUM} + \mathrm{Q_8H_2} \tag{70}$$

$$\mathrm{SUCC} + \mathrm{Q_8} \xrightleftharpoons{\mathrm{SUCDI}} \mathrm{FUM} + \mathrm{Q_8H_2} \tag{71}$$

$$\mathrm{FUM} + \mathrm{H_2O} \xrightleftharpoons{\mathrm{FUM}} \mathrm{MALL} \tag{72}$$

$$\mathrm{MALL} + \mathrm{NAD} \xrightleftharpoons{\mathrm{MDH}} \mathrm{OAA} + \mathrm{H} + \mathrm{NADH} \tag{73}$$

$$\mathrm{NADPH} + \mathrm{NAD} \xrightleftharpoons{\mathrm{NADTRHD}} \mathrm{NADP} + \mathrm{NADH} \tag{74}$$

$$\mathrm{COA} + \mathrm{PYR} + \mathrm{NAD} \xrightleftharpoons{\mathrm{PDH}} \mathrm{ACCOA} + \mathrm{CO_2} + \mathrm{NADH} \tag{75}$$

$$\mathrm{COA} + \mathrm{PYR} \xrightleftharpoons{\mathrm{PFL}} \mathrm{ACCOA} + \mathrm{FOR} \tag{76}$$

$$4\,\mathrm{H} + \mathrm{NADH} + \mathrm{Q_8} \xrightleftharpoons{\mathrm{NADH_{16}}} 3\,\mathrm{H_E} + \mathrm{NAD} + \mathrm{Q_8H_2} \tag{77}$$

$$\mathrm{O_2} + 4\,\mathrm{H} + 2\,\mathrm{Q_8H_2} \xrightleftharpoons{\mathrm{CYTBD}} 4\,\mathrm{H_E} + 2\,\mathrm{H_2O} + 2\,\mathrm{Q_8} \tag{78}$$

$$\mathrm{ADP} + \mathrm{PI} + 4\,\mathrm{H_E} \xrightleftharpoons{\mathrm{ATPS_4R}} \mathrm{ATP} + 3\,\mathrm{H} + \mathrm{H_2O} \tag{79}$$

[34]:
```
chemostats0 = ['ADP','ATP','H2O','PI','H']
chemostats = ['GLCD_E','CO2','O2']
chemostats.extend(chemostats0)
sc = st.statify(s,chemostats=chemostats)
print('Chemostats:', printChem(chemostats))
print(st.sprintp(sc))
```

```
Chemostats: \ch{GLCD_E}, \ch{CO2}, \ch{O2}, \ch{ADP}, \ch{ATP}, \ch{H2O},
\ch{PI}, \ch{H}
3 pathways
0:  + PFK + FBP
1:  - FRD7 + SUCDI
2:  + 2 GLCPTS + 2 PGI + 2 PFK + 2 FBA + 2 TPI + 4 GAPD + 4 PGK + 4 PGM + 4 ENO
+ 2 PYK + 4 CS + 4 ACONTA + 4 ACONTB + 4 ICDHYR + 4 AKGDH + 4 SUCOAS + 4 FRD7 +
4 FUM + 4 MDH + 4 NADTRHD + 4 PDH + 20 NADH16 + 12 CYTBD + 27 ATPS4R
```

[35]:
```
sp = st.path(s,sc)
disp.Latex(st.sprintrl(sp,chemformula=True))
```

[35]:

$$\mathrm{ATP} + \mathrm{H_2O} \overset{\mathrm{pr}_1}{\rightleftharpoons} \mathrm{ADP} + \mathrm{PI} + \mathrm{H} \tag{80}$$

$$\overset{\mathrm{pr}_2}{\rightleftharpoons} \tag{81}$$

$$2\,\mathrm{GLCD_E} + 12\,\mathrm{O_2} + 35\,\mathrm{ADP} + 35\,\mathrm{PI} + 35\,\mathrm{H} \overset{\mathrm{pr}_3}{\rightleftharpoons} 12\,\mathrm{CO_2} + 35\,\mathrm{ATP} + 47\,\mathrm{H_2O} \tag{82}$$

There are three pathways:

- A futile cycle
- A null cycle
- Each GLCD_E, combined with $6\,O_2$, reverses ATP hydrolysis ($ATP + H_2O \rightleftharpoons ADP + PI + H^+$) to give 17.5 ATP molecules with an additional $6\,H_2O$ and $6\,CO_2$; 17.5 ATP is the value quoted by \cite{{Pal15}}.

[ ]:

References

F.E. Cellier and J. Greifeneder. Modeling chemical reactions in modelica by use of chemo-bonds. In *Proceedings 7th Modelica Conference*, Como, Italy, September 2009.

Reginald H. Garrett and Charles M. Grisham. *Biochemistry*. Cengage Learning, Boston, MA, 6th edition, 2017.

P. J. Gawthrop. Bond graph modeling of chemiosmotic biomolecular energy transduction. *IEEE Transactions on NanoBioscience*, 16(3):177–188, April 2017a. ISSN 1536-1241. doi: 10.1109/TNB.2017.2674683. Available at arXiv:1611.04264.

P. J. Gawthrop and E. J. Crampin. Modular bond-graph modelling and analysis of biomolecular systems. *IET Systems Biology*, 10(5):187–201, October 2016. ISSN 1751-8849. doi: 10.1049/iet-syb.2015.0083. Available at arXiv:1511.06482.

P. J. Gawthrop, I. Siekmann, T. Kameneva, S. Saha, M. R. Ibbotson, and E. J. Crampin. Bond graph modelling of chemoelectrical energy transduction. *IET Systems Biology*, 11(5):127–138, 2017. ISSN 1751-8849. doi: 10.1049/iet-syb.2017.0006. Available at arXiv:1512.00956.

Peter J. Gawthrop. Bond-graph modelling and causal analysis of biomolecular systems. In Wolfgang Borutzky, editor, *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*, pages 587–623. Springer International Publishing, Berlin, 2017b. ISBN 978-3-319-47434-2. doi: 10.1007/978-3-319-47434-2_16.

Peter J. Gawthrop and Edmund J. Crampin. Energy-based analysis of biochemical cycles using bond graphs. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 470(2171):1–25, 2014. doi: 10.1098/rspa.2014.0459. Available at arXiv:1406.2447.

Peter J. Gawthrop and Edmund J. Crampin. Energy-based analysis of biomolecular pathways. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 473 (2202), 2017. ISSN 1364-5021. doi: 10.1098/rspa.2016.0825. Available at arXiv:1611.02332.

Peter J. Gawthrop and Edmund J. Crampin. Biomolecular system energetics. In *Proceedings of the 13th International Conference on Bond Graph Modeling (ICBGM'18)*, Bordeaux, 2018. Society for Computer Simulation. Available at arXiv:1803.09231.

Peter J. Gawthrop, Joseph Cursons, and Edmund J. Crampin. Hierarchical bond graph modelling of biochemical networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2184):1–23, 2015. ISSN 1364-5021. doi: 10.1098/rspa.2015.0642. Available at arXiv:1503.01814.

J. Orth, R. Fleming, and B. Palsson. Reconstruction and use of microbial metabolic networks: the core escherichia coli metabolic model as an educational guide. *EcoSal Plus*, 2010a. doi: 10.1128/ecosalplus.10.2.1.

Jeffrey D. Orth, Ines Thiele, and Bernhard O. Palsson. What is flux balance analysis? *Nat Biotech*, 28:245–248, March 2010b. ISSN 1087-0156. doi: 10.1038/nbt.1614.

George Oster, Alan Perelson, and Aharon Katchalsky. Network thermodynamics. *Nature*, 234: 393–399, December 1971. doi: 10.1038/234393a0.

George F. Oster, Alan S. Perelson, and Aharon Katchalsky. Network thermodynamics: dynamic modelling of biophysical systems. *Quarterly Reviews of Biophysics*, 6(01):1–134, 1973. doi: 10.1017/S0033583500000081.

Bernhard Palsson. *Systems biology: properties of reconstructed networks*. Cambridge University Press, 2006. ISBN 0521859034.

Bernhard Palsson. *Systems Biology: Simulation of Dynamic Network States*. Cambridge University Press, 2011.

Bernhard Palsson. *Systems Biology: Constraint-Based Reconstruction and Analysis*. Cambridge University Press, Cambridge, 2015.

Michael Pan, Peter J. Gawthrop, Kenneth Tran, Joseph Cursons, and Edmund J. Crampin. A thermodynamic framework for modelling membrane transporters. *Journal of Theoretical Biology*, 481:10 – 23, 2019. ISSN 0022-5193. doi: 10.1016/j.jtbi.2018.09.034. Available at arXiv:1806.04341.