Redox

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

December 15, 2020

Contents

1 Introduction **1**

2 Redox reaction **2**
2.1 Bond graph 2
2.2 Stoichiometry 3
2.3 Reactions 3
2.4 Potentials 3

3 Proton pump **4**
3.1 Bond graph 4
3.2 Stoichiometry 6
3.3 Reactions 6

4 Complex I **6**
4.1 Stoichiometry 7
4.2 Reactions 7

5 Superoxide generation and removal (additional material) **8**
5.1 Superoxide dismutase (MnSOD) 8
5.2 Glutathione peroxidase (GPx1) 10

Note:

- This example is discussed in detail by Gawthrop and Pan [2020] available here.
- This is the Redox.ipynb notebook. The PDF version is available here.

1 Introduction

Redox reactions and proton pumps play a fundamental role in Biology. This note illustrates this using a bond graph model of complex I of the mitochondrial electron transport chain.

[1]:
```
## Some useful imports
import BondGraphTools as bgt
import numpy as np
import sympy as sp
import matplotlib.pyplot as plt

## Stoichiometric analysis
import stoich as st
```

```
## SVG
import svgBondGraph as sbg

## Display (eg disp.SVG(), disp.
import IPython.display as disp

## Potential data
import phiData

quiet = True
```

2 Redox reaction

A key energy-generating redox reaction that within the mitochondrial respiratory chain is

$$NADH + Q + H^+ \overset{r}{\rightleftharpoons} NAD^+ + QH_2 \tag{1}$$

This reaction can be divided into the half reactions:

$$NADH \overset{r_1}{\rightleftharpoons} 2\,e_1^- + H^+ + NAD^+ \tag{2}$$

$$2\,e_2^- + 2\,H^+ + Q \overset{r_2}{\rightleftharpoons} QH_2 \tag{3}$$

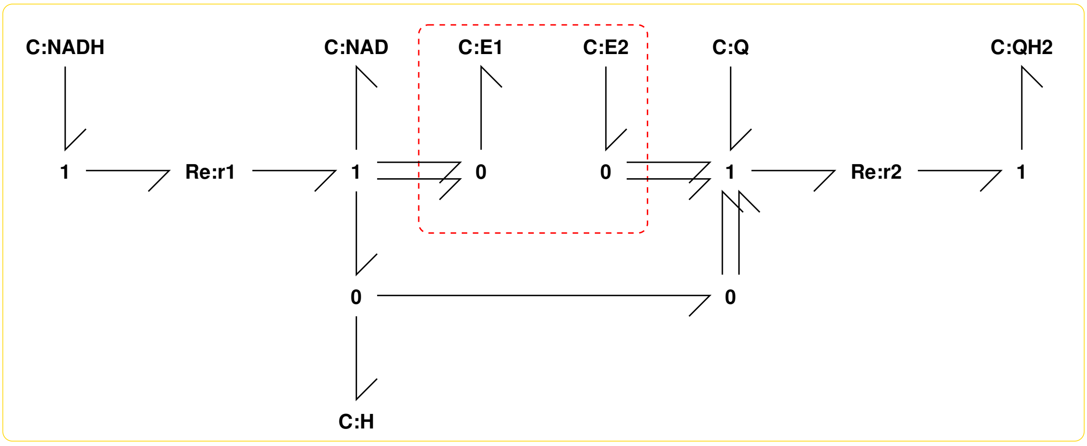
A bond graph representation of this decomposition is given below.

2.1 Bond graph

C:E1 and C:E2 represent the electron potentials and the other The C components represent the species; the two Re components the two half reactions.

[2]:
```
## Redox reaction
sbg.model('Redox_abg.svg')
import Redox_abg
disp.SVG('Redox_abg.svg')
```

[2]:



2.2 Stoichiometry

[3]:
```
## Stoichiometry
linear = ['E1','E2']
s = st.stoich(Redox_abg.model(),linear=linear,quiet=quiet)
chemostats = ['NADH','NAD','Q','QH2','H']
sc = st.statify(s,chemostats=chemostats)
```

[4]:
```
## Stoichiometric matrix
disp.Latex(st.sprintl(s,'N'))
```

[4]:

$$N = \begin{pmatrix} 2 & 0 \\ 0 & -2 \\ 1 & -2 \\ 1 & 0 \\ -1 & 0 \\ 0 & -1 \\ 0 & 1 \end{pmatrix} \tag{4}$$

2.3 Reactions

These are automatically generated from the bond graph

[5]:
```
## Reactions
disp.Latex(st.sprintrl(s,chemformula=True,all=True))
```

[5]:

$$\mathrm{NADH} \xrightleftharpoons{r_1} 2\,\mathrm{E_1} + \mathrm{H} + \mathrm{NAD} \tag{5}$$

$$2\,\mathrm{E_2} + 2\,\mathrm{H} + \mathrm{Q} \xrightleftharpoons{r_2} \mathrm{QH_2} \tag{6}$$

2.4 Potentials

The reaction (Faraday Equivalent) potentials are computed from tables gleaned from the literature

[6]:
```
## Standard potetials
phi_Std = phiData.phi_Std()

## Typical concentrations
conc = phiData.ParRubXu16_conc()

## From BazBeaVin16
conc['Q'] = conc['QH2'] = 1e-2

## pH 7
conc['H'] = 1e-7

## Table for paper and put values in to phi_NADH etc.
print('%% Table')
ch='\ch'
```

```
l='{'
r='}'
eol = r'\\'
phi_std = {}
for spec in ['NAD','NADH','Q','QH2','H']:
    phi0 = phi_Std[spec]
    con = conc[spec]
    phi_std_spec = phi0 + st.V_N()*np.log(con)
    phi_std[spec] = phi_std_spec
    #print(f'phi_Std_{spec} = {1000*phi0:0.0f}, phi_{spec} =␣
 ↪{1000*phi_std[spec]:.0f}, conc_{spec}={conc[spec]}')
    print(f'{ch}{l}{spec}{r} & {1000*phi0:.0f} & {con:1.2e} &␣
 ↪{1000*phi_std_spec:.0f}{eol}')
    exec(f'phi_{spec} = {phi_std_spec}')


## Print the worked example for the paper.
print('\n%% Equations')
E1 = 0.5*(phi_NADH - phi_NAD - phi_H)
E2 = 0.5*(phi_QH2 - phi_Q - 2*phi_H)
print(f'E1 = 0.5({1000*phi_NADH:.0f} - {1000*phi_NAD:.0f} - {1000*phi_H:.0f})␣
 ↪= {1000*E1:.0f} mV')
print(f'E2 = 0.5({1000*phi_QH2:.0f} - {1000*phi_Q:.0f} - 2x{1000*phi_H:.0f})␣
 ↪= {1000*E2:.0f} mV')
print(f'E1-E2 = {1000*(E1-E2):.0f} mV')
print(f'PMF = {1000*(E1-E2)/2:.0f} mV')
```

```
%% Table
\ch{NAD} & 188 & 5.02e-04 & -15\\
\ch{NADH} & 407 & 7.50e-05 & 154\\
\ch{Q} & 675 & 1.00e-02 & 552\\
\ch{QH2} & -241 & 1.00e-02 & -365\\
\ch{H} & 0 & 1.00e-07 & -431\\

%% Equations
E1 = 0.5(154 - -15 - -431) = 300 mV
E2 = 0.5(-365 - 552 - 2x-431) = -28 mV
E1-E2 = 328 mV
PMF = 164 mV
```

3 Proton pump

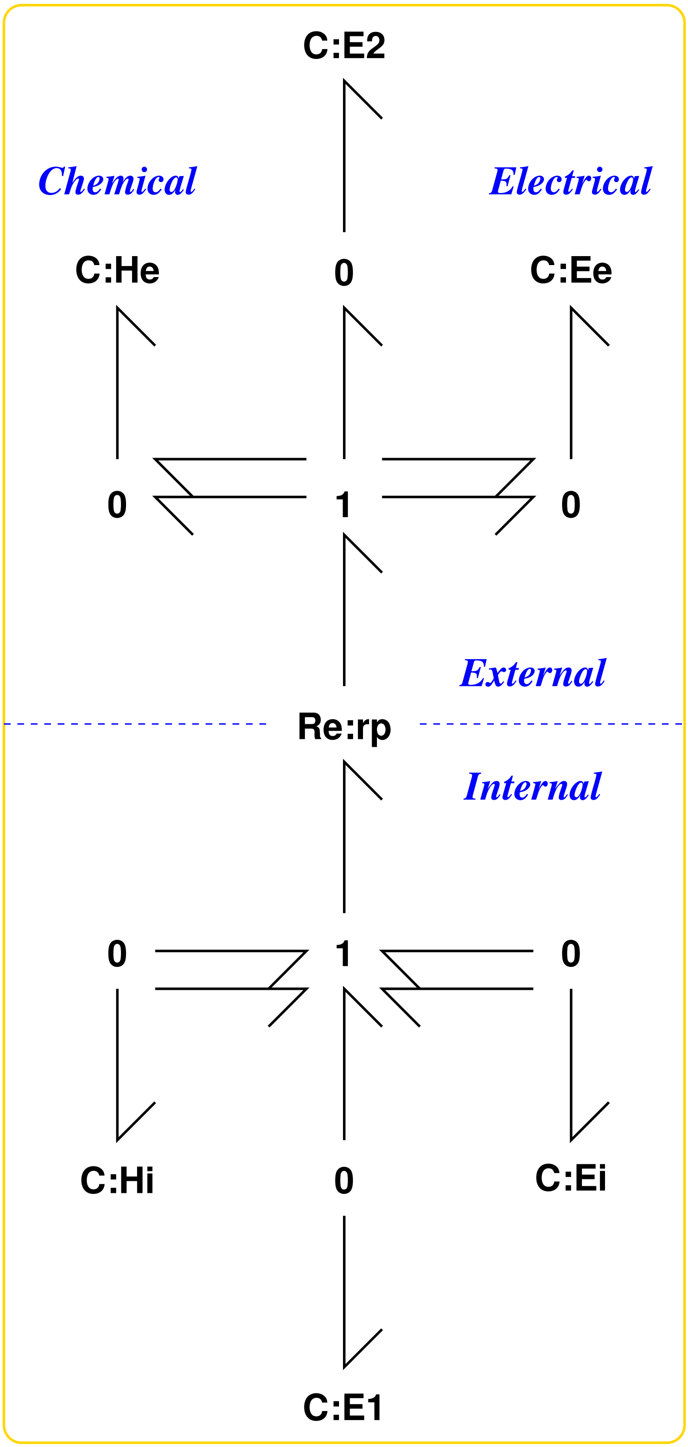
The redox reaction of complex I drives a proton pump.

3.1 Bond graph

C:E1 and C:E2 correspond to the redox reaction and provide the potential to drive protons in the interior H_i^+ to the exterior H_e^+ of the mitochondrial membrane. The protons have both electrical and chemoical potential.

[7]:
```
## Proton pump
sbg.model('ProtonPump_abg.svg')
import ProtonPump_abg
disp.SVG('ProtonPump_abg.svg')
```

[7]:



3.2 Stoichiometry

[8]:
```
## Stoichiometry
linear = ['E1','E2','Ei','Ee']
s = st.stoich(ProtonPump_abg.model(),linear=linear,quiet=quiet)
chemostats = ['E1','E2','Ei','Ee','Hi','He']
sc = st.statify(s,chemostats=chemostats)
#print(s['species'])
#disp.Latex(st.sprint(s0,'K'))
#print(st.sprints(s))
```

3.3 Reactions

These are automatically generated from the bond graph

[9]:
```
## Reactions
disp.Latex(st.sprintrl(s,chemformula=True,all=True))
```

[9]:

$$\mathrm{E_1} + 2\,\mathrm{Ei} + 2\,\mathrm{Hi} \xrightleftharpoons{\mathrm{rp}} \mathrm{E_2} + 2\,\mathrm{Ee} + 2\,\mathrm{He} \tag{7}$$

[10]:
```
## Flows
disp.Latex(st.sprintvl(s))
```

[10]:

$$v_{rp} = \kappa_{rp}\left(-K_{He}^2 x_{He}^2 e^{\frac{K_{E2}x_{E2}+2K_{Ee}x_{Ee}}{V_N}} + K_{Hi}^2 x_{Hi}^2 e^{\frac{K_{E1}x_{E1}+2K_{Ei}x_{Ei}}{V_N}}\right) \tag{8}$$

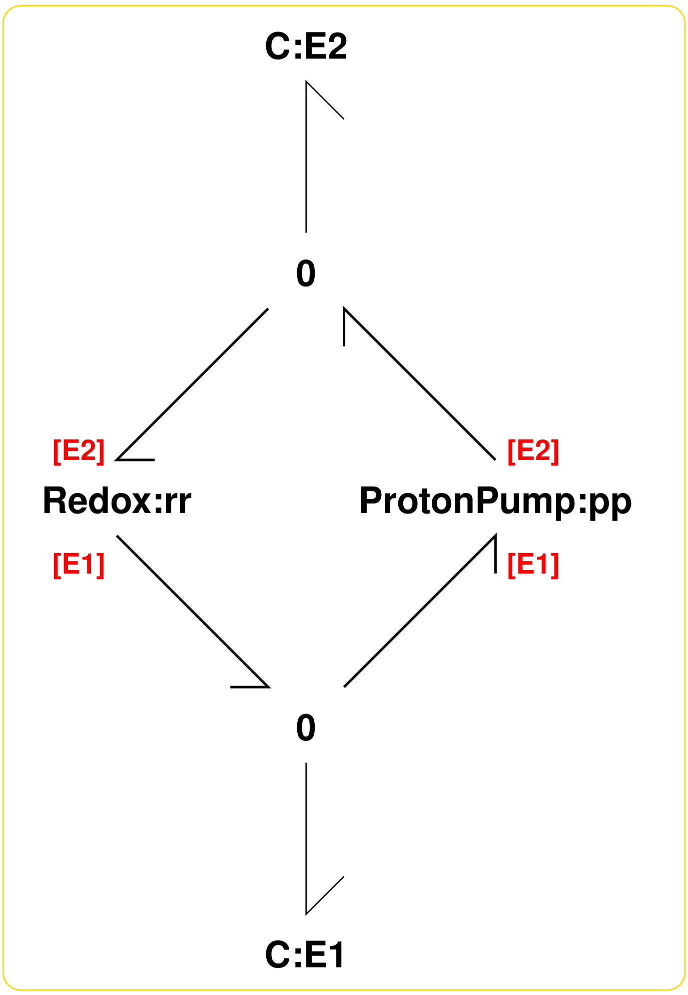
4 Complex I

The model of mitochondrial complex I comprides tow modules: the redox reaction and the proton pump.

[11]:
```
## Complex I
sbg.model('ComplexI_abg.svg')
import ComplexI_abg
disp.SVG('ComplexI_abg.svg')
```

```
Creating subsystem: ProtonPump:pp
Creating subsystem: Redox:rr
```

[11]:



4.1 Stoichiometry

[12]:
```python
## Stoichiometry
linear = ['E1','E2','pp_Ei','pp_Ee']
s = st.stoich(ComplexI_abg.model(),linear=linear,quiet=quiet)
print(s['species'])
chemostats = ['pp_Ee', 'pp_Ei', 'pp_He', 'pp_Hi', 'rr_H', 'rr_NAD',␣
 ↪'rr_NADH', 'rr_Q', 'rr_QH2']
sc = st.statify(s,chemostats=chemostats)
#print(s['species'])
#disp.Latex(st.sprint(s0,'K'))
#print(st.sprints(s))
```

```
['E1', 'E2', 'pp_Ee', 'pp_Ei', 'pp_He', 'pp_Hi', 'rr_H', 'rr_NAD', 'rr_NADH',
'rr_Q', 'rr_QH2']
```

4.2 Reactions

These are automatically generated from the bond graph

[13]:
```
## Reactions
disp.Latex(st.sprintrl(s,chemformula=True,all=True))
```
[13]:

$$\mathrm{E_1 + 2\,pp_Ei + 2\,pp_Hi} \xrightleftharpoons{\mathrm{pp_rp}} \mathrm{E_2 + 2\,pp_Ee + 2\,pp_He} \tag{9}$$
$$\mathrm{rr_NADH} \xrightleftharpoons{\mathrm{rr}} \mathrm{2\,E_1 + rr_H + rr_NAD} \tag{10}$$
$$\mathrm{2\,E_2 + 2\,rr_H + rr_Q} \xrightleftharpoons{\mathrm{rr}} \mathrm{rr_QH_2} \tag{11}$$

[14]:
```
## Flows
disp.Latex(st.sprintvl(s))
```
[14]:

$$v_{pprp} = \kappa_{pprp}\left(-K_{ppHe}^2 x_{ppHe}^2 e^{\frac{K_{E2}x_{E2}+2K_{ppEe}x_{ppEe}}{V_N}} + K_{ppHi}^2 x_{ppHi}^2 e^{\frac{K_{E1}x_{E1}+2K_{ppEi}x_{ppEi}}{V_N}}\right) \tag{12}$$
$$v_{rr} = \kappa_{rr}\left(-K_{rrH}K_{rrNAD}x_{rrH}x_{rrNAD}e^{\frac{2K_{E1}x_{E1}}{V_N}} + K_{rrNADH}x_{rrNADH}\right) \tag{13}$$
$$v_{rr} = \kappa_{rr}\left(K_{rrH}^2 K_{rrQ} x_{rrH}^2 x_{rrQ} e^{\frac{2K_{E2}x_{E2}}{V_N}} - K_{rrQH2}x_{rrQH2}\right) \tag{14}$$

[15]:
```
## Path
sp = st.path(s,sc)
## Reactions
disp.Latex(st.sprintrl(sp,chemformula=True,all=True))
```
[15]:

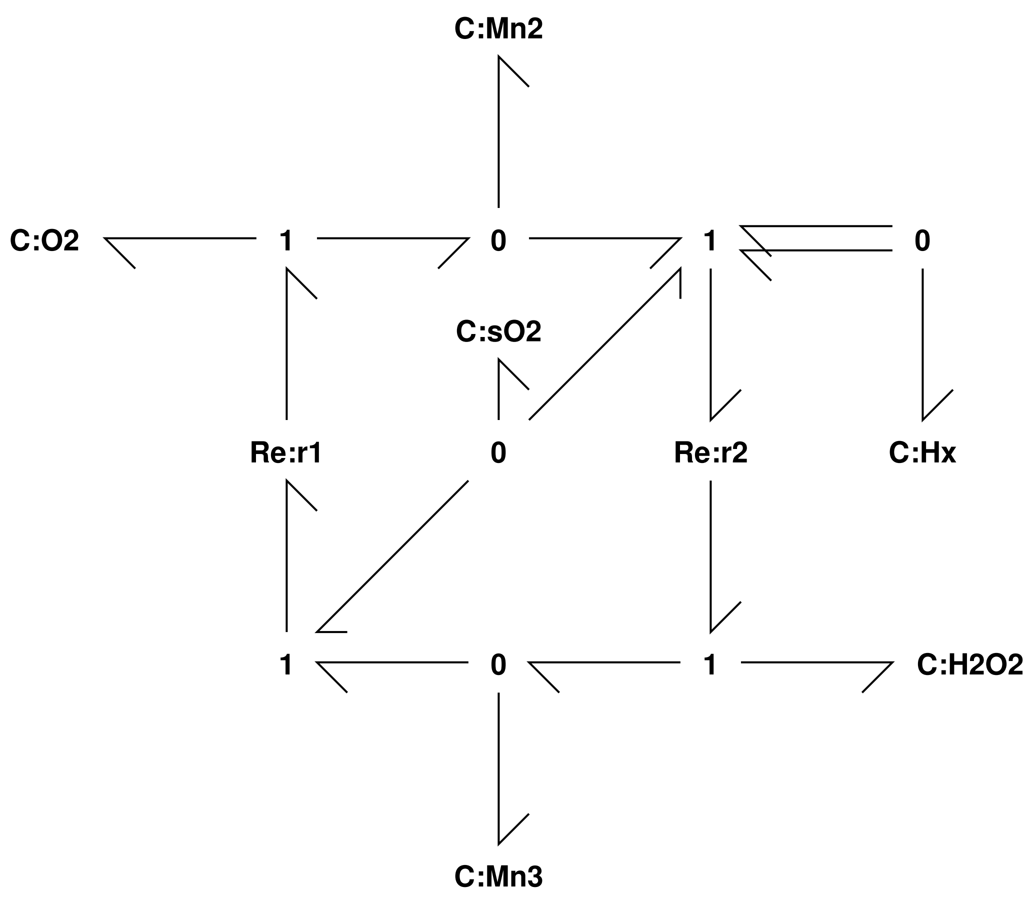
$$\mathrm{4\,pp_Ei + 4\,pp_Hi + rr_H + rr_NADH + rr_Q} \xrightleftharpoons{\mathrm{pr_1}} \mathrm{4\,pp_Ee + 4\,pp_He + rr_NAD + rr_QH_2} \tag{15}$$

5 Superoxide generation and removal (additional material)

Superoxide is generated from the electron potential of the redox reaction and free oxygen by the reaction: $\mathrm{e^- + O_2 \rightleftharpoons O_2^{\bullet -}}$ It is removed by superoxide dismutase in conjunction with glutathione peroxidase. THe bond graph representation of these mechanisms are given below.

5.1 Superoxide dismutase (MnSOD)

[16]:
```
## Complex I
sbg.model('MnSOD_abg.svg')
import MnSOD_abg
disp.SVG('MnSOD_abg.svg')
```
[16]:



[17]:
```
## Stoichiometry
s = st.stoich(MnSOD_abg.model(),quiet=quiet)
chemostats = ['H2O2', 'Hx', 'O2', 's02']
sc = st.statify(s,chemostats=chemostats)
#disp.Latex(st.sprint(s0,'K'))
#print(st.sprints(s))
```

[18]:
```
## Reactions
disp.Latex(st.sprintrl(s,chemformula=True,all=True))
```

[18]:

$$Mn_3 + sO_2 \xrightleftharpoons{r_1} Mn_2 + O_2 \tag{16}$$

$$2\,Hx + Mn_2 + sO_2 \xrightleftharpoons{r_2} H_2O_2 + Mn_3 \tag{17}$$

[19]:
```
## Flows
disp.Latex(st.sprintvl(s))
```

[19]:

$$v_{r1} = \kappa_{r1}\left(-K_{Mn2}K_{O2}x_{Mn2}x_{O2} + K_{Mn3}K_{sO2}x_{Mn3}x_{sO2}\right) \tag{18}$$

$$v_{r2} = \kappa_{r2}\left(-K_{H2O2}K_{Mn3}x_{H2O2}x_{Mn3} + K_{Hx}^2K_{Mn2}K_{sO2}x_{Hx}^2x_{Mn2}x_{sO2}\right) \tag{19}$$

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

[20]:

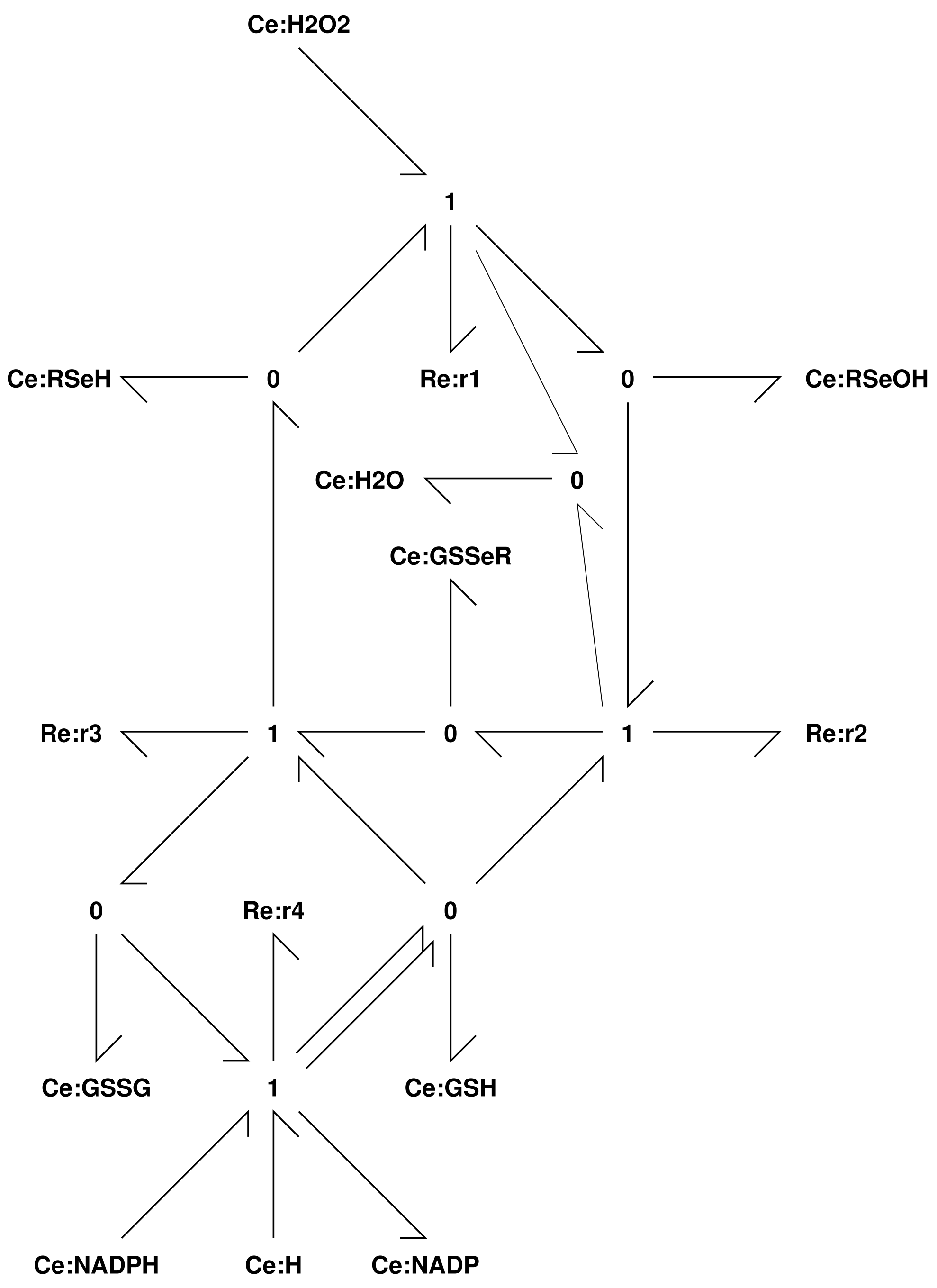
$$2\,Hx + 2\,sO_2 \xrightleftharpoons{pr_1} H_2O_2 + O_2 \tag{20}$$

5.2 Glutathione peroxidase (GPx1)

```
[21]: ## Complex I
      sbg.model('GPx1_abg.svg')
      import GPx1_abg
      disp.SVG('GPx1_abg.svg')
```

```
Converting one-port r1 to two-port
Converting one-port r4 to two-port
Converting one-port r2 to two-port
Converting one-port r3 to two-port
```

[21]:



```
[22]: ## Stoichiometry
      s = st.stoich(GPx1_abg.model(),quiet=quiet)
      #print(s['species'])
      chemostats = ['GSH', 'GSSG', 'H', 'H2O', 'H2O2','NADP','NADPH']
      sc = st.statify(s,chemostats=chemostats)
      #disp.Latex(st.sprint(s0,'K'))
      #print(st.sprints(s))
```

```
[23]: ## Reactions
      disp.Latex(st.sprintrl(s,chemformula=True,all=True))
```

[23]:

$$H_2O_2 + RSeH \xrightleftharpoons{r_1} H_2O + RSeOH \tag{21}$$
$$GSH + RSeOH \xrightleftharpoons{r_2} GSSeR + H_2O \tag{22}$$
$$GSH + GSSeR \xrightleftharpoons{r_3} GSSG + RSeH \tag{23}$$
$$GSSG + H + NADPH \xrightleftharpoons{r_4} 2\,GSH + NADP \tag{24}$$

[24]:
```
## Flows
disp.Latex(st.sprintvl(s))
```

[24]:

$$v_{r1} = \kappa_{r1}\left(-K_{H2O}K_{RSeOH}x_{H2O}x_{RSeOH} + K_{H2O2}K_{RSeH}x_{H2O2}x_{RSeH}\right) \tag{25}$$
$$v_{r2} = \kappa_{r2}\left(K_{GSH}K_{RSeOH}x_{GSH}x_{RSeOH} - K_{GSSeR}K_{H2O}x_{GSSeR}x_{H2O}\right) \tag{26}$$
$$v_{r3} = \kappa_{r3}\left(K_{GSH}K_{GSSeR}x_{GSH}x_{GSSeR} - K_{GSSG}K_{RSeH}x_{GSSG}x_{RSeH}\right) \tag{27}$$
$$v_{r4} = \kappa_{r4}\left(-K_{GSH}^2K_{NADP}x_{GSH}^2x_{NADP} + K_{GSSG}K_HK_{NADPH}x_{GSSG}x_Hx_{NADPH}\right) \tag{28}$$

[25]:
```
sp = st.path(s,sc)
## Reactions
disp.Latex(st.sprintrl(sp,chemformula=True,all=True))
```

[25]:

$$2\,GSH + H_2O_2 \xrightleftharpoons{pr_1} GSSG + 2\,H_2O \tag{29}$$

[ ]:

[ ]:

[ ]:

References

Peter J. Gawthrop and Michael Pan. Network thermodynamical modelling of bioelectrical systems: A bond graph approach. Available at arXiv:2009.02217, 2020.