In [1]:
```
import mfapy
%matplotlib inline
import numpy as np
import scipy.integrate
from matplotlib import pyplot as plt
```

In [2]:
```
# User input code goes here:

simulation_model = input("Enter the name of the metabolic model .txt file: ")

simulation_csv = input("Enter the name of the .csv file containing the parame

MDV_data = input("Enter the name of the MDV .txt file: ")
```

```
Enter the name of the metabolic model .txt file: Elim2_model.txt
Enter the name of the .csv file containing the parameters: Elim2_status.csv
Enter the name of the MDV .txt file: Elim2_mdv.txt
```

In [3]:
```
#Sets up model variables using simulation model.txt file
reactions, reversible, metabolites, target_fragments = mfapy.mfapyio.load_met
model = mfapy.metabolicmodel.MetabolicModel(reactions, reversible, metabolite

#loads states and values from .csv file
flux = model.load_states(simulation_csv, format = 'csv')
model.set_constraints_from_state_dict(flux)
model.update()
```

```
Caution: Formate was not used in the metabolid network
```

Out[3]:
```
True
```

In [4]:
```
# Generation of CarbonSource instance
cs1 = model.generate_carbon_source_template()
cs1.set_each_isotopomer('SubsCO2',{'#0': 0.01, '#1': 0.99}, correction = 'yes
cs1.set_each_isotopomer('Subsmeoh',{'#0': 1.0, '#1': 0.0}, correction = 'yes'
```

Out[4]:
```
True
```

In [5]:
```
# Load MDV data
mdv1 = model.load_mdv_data(MDV_data) #string variable from user input

#include more if there are more data files
```

In [6]:
```
# Flux estimation Step 1: Setting experimments
model.set_experiment('ex1', mdv1, cs1)
#include more for more data sets
```

Out[6]:
```
True
```

In [7]:
```
# Flux estimation step 2: Generation of intical flux vectors
initialstate, flux_initial = model.generate_initial_states(50, 4, method ="nc
```

In [8]:
```
# Flux estimation step 3: Fitting model

#GN_CRS2_LM
initialstate, RSS, flux_opt_GN_CRS2_LM = model.fitting_flux(method = "GN_CRS2

#SLSQP
initialstate, RSS, flux_opt_SLSQP = model.fitting_flux(method = "SLSQP", flux

#LN_SBPLX
initialstate, RSS, flux_opt_LN_SBPLX = model.fitting_flux(method = "LN_SBPLX"
```

In [9]:
```
#Print results of each method onto the console
model.show_results([("GN_CRS2_LM", flux_opt_GN_CRS2_LM[0])])
```

```
1.0   0.01000.0 ABCDE-->ABCDE
r62             E4P+PEP+PEP-->Phe        (kegg:Biom    10.0 pseudo     1.0
1.0   0.01000.0 ABCD+EFG+HIJ-->EFGIJABCD
r63             E4P+PEP+PEP-->Tyr        (kegg:Biom    10.0 pseudo     1.0
1.0   0.01000.0 ABCD+EFG+HIJ-->EFGIJABCD
r64             Oxa+Pyr-->Ile            (kegg:R015    10.0 pseudo     1.0
1.0   0.01000.0 ABCD+EFG-->ABFGCD
r65             AcCOA+Pyr+Pyr-->Leu      (kegg:Biom    10.0 pseudo     1.0
1.0   0.01000.0 AB+CDE+FGH-->ABDEGH
r66             Pyr+Pyr-->Val            (kegg:Biom    10.0 pseudo     1.0
1.0   0.01000.0 ABC+DEF-->ABECF
r67             PGA-->Ser                (kegg:Biom     4.3 free       1.0
1.0   0.01000.0 ABC-->ABC
r68             Ser-->Gly+MEETHF         (kegg:Glu2   773.8 free       1.0
1.0   0.01000.0 ABC-->AB+C
r69             Gly+MEETHF-->Ser         (kegg:Biom   771.1 free       1.0
1.0   0.01000.0 AB+C-->ABC
r72             G6P-->Biomass            (kegg:Biom     1.0 fixed      1.0
1.0   0.01000.0 nd
```

In [10]:
```
model.show_results([("SLSQP", flux_opt_SLSQP[0])])
```

```
Id              Reaction                 External i   SLSQP
RSS                                                462854.64
Thres                                                 38.89
p_value                                                0.00
Id              Reaction                 External i   SLSQP Type     Value
Stdev    lb     ub Atom_mapping
r1              SubsGlc-->G6P            (kegg:R028     0.0 fixed      0.0
1.0   0.01000.0 ABCDEF-->ABCDEF
r2              G6P-->F6P                (kegg:R007   259.6 free       1.0
1.0   0.01000.0 ABCDEF-->ABCDEF
r3              F6P-->G6P                (kegg:R007   272.1 free       1.0
1.0   0.01000.0 ABCDEF-->ABCDEF
r4              F6P-->FBP                (kegg:R007   791.7 free       1.0
1.0   0.01000.0 ABCDEF-->ABCDEF
r5              FBP-->F6P                (kegg:R010   800.3 free       1.0
1.0   0.01000.0 ABCDEF-->ABCDEF
r6              FBP-->DHAP+GAP           (kegg:R010   567.2 free       1.0
1.0   0.01000.0 ABCDEF-->CBA+DEF
r7              DHAP+GAP-->FBP           (kegg:R010   575.7 free       1.0
```

In [11]:
```
model.show_results([("LN_SBPLX", flux_opt_LN_SBPLX[0])])
```

```
Id              Reaction                  External i LN_SBPL
RSS                                                56894.39
Thres                                                 38.89
p_value                                                0.00
Id              Reaction                  External i LN_SBPL Type     Value
Stdev    lb     ub Atom_mapping
r1              SubsGlc-->G6P             (kegg:R028     0.0 fixed      0.0
1.0   0.01000.0 ABCDEF-->ABCDEF
r2              G6P-->F6P                 (kegg:R007   544.0 free       1.0
1.0   0.01000.0 ABCDEF-->ABCDEF
r3              F6P-->G6P                 (kegg:R007   551.0 free       1.0
1.0   0.01000.0 ABCDEF-->ABCDEF
r4              F6P-->FBP                 (kegg:R007   991.5 free       1.0
1.0   0.01000.0 ABCDEF-->ABCDEF
r5              FBP-->F6P                 (kegg:R010  1000.0 free       1.0
1.0   0.01000.0 ABCDEF-->ABCDEF
r6              FBP-->DHAP+GAP            (kegg:R010   991.5 free       1.0
1.0   0.01000.0 ABCDEF-->CBA+DEF
r7              DHAP+GAP-->FBP            (kegg:R010  1000.0 free       1.0
```

In [12]:
```
# Output fitted metabolic flux into CSV file

model.show_results([("GN_CRS2_LM", flux_opt_GN_CRS2_LM[0])], filename = "Elim

model.show_results([("SLSQP", flux_opt_SLSQP[0])], filename = "Elim2_output_S

model.show_results([("LN_SBPLX", flux_opt_LN_SBPLX[0])], filename = "Elim2_ou
```