**E. coli (iJO1366) "Valine, Leucine, and Isoleucine Metabolism"**

Author: H. Scott Hinton, Utah State University

Reviewer:

**INTRODUCTION**

The purpose of this tutorial is to review the basic structure and capabilities of the "Valine, Leucine, and Isoleucine Metabolism" subsystem of the iJO1366 E.coli model.

**MATERIALS**

This tutorial is based on the Constraint-Based Reconstruction and Analysis (COBRA) Toolbox [1,2]. To use this tutorial requires the 2016a or newer version of Matlab (https://www.mathworks.com/) and the COBRA toolbox software that can be downloaded from https://opencobra.github.io/cobratoolbox/latest/index.html. The installation instructions and troubleshooting tips are also available on this website.

**EQUIPMENT SETUP**

Initialize the COBRA toolbox.

```plaintext
initCobraToolbox
```

> Checking if git is installed ...  Done.
> Checking if the repository is tracked using git ...  Done.
> Checking if curl is installed ...  Done.
> Checking if remote can be reached ...  Done.
> Initializing and updating submodules ...  Done.
> Adding all the files of The COBRA Toolbox ...  Done.
> Define CB map output... set to svg.
> Retrieve models ...  Done.
> TranslateSBML is installed and working properly.
> Configuring solver environment variables ...
  - [---] ILOG_CPLEX_PATH :  --> set this path manually after installing the solver ( see instructions )
  - [---] Gurobi_PATH: C:\gurobi17.0\win64\matlab
  - [---] TOMLAB_PATH :  --> set this path manually after installing the solver ( see instructions )
  - [---] MOSEK_PATH :  --> set this path manually after installing the solver ( see instructions )
  Done.
> Checking available solvers and solver interfaces ...  Done.
> Setting default solvers ...  Done.
> Saving the MATLAB path ...  Done.
> The MATLAB path was saved in the default location.

> Summary of available solvers and solver interfaces

<table>
<thead>
<tr>
<th>Support</th>
<th>LP</th>
<th>MILP</th>
<th>QP</th>
<th>MIQP</th>
<th>NLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>cplex_direct</td>
<td>full</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
dqqMinos full 0 - - - -
glpk full 1 1 - - -
gurobi full 1 1 1 1 -
lp_m_cplex full 0 0 0 0 -
matlab full 1 - - - 1
mosek full 0 0 0 - -
pdco full 1 - 1 - 1
quadMinos full 0 - - - 0
tomlab_cplex full 0 0 0 0 -
qpng experimental - - 1 - -
tomlab_snopt experimental - - - - 0
gurobi_mex legacy 0 0 0 0 -
lindo_old legacy 0 - - - -
lindo_legacy legacy 0 - - - -
lp_solve legacy 1 - - - -
opti legacy 0 0 0 0 0

Total - 5 2 3 1 2

+ Legend: = not applicable, @ = solver not compatible or not installed, 1 = solver installed.

> You can solve LP problems using: 'glpk' - 'gurobi' - 'matlab' - 'pdco' - 'lp_solve'
> You can solve MILP problems using: 'glpk' - 'gurobi'
> You can solve QP problems using: 'gurobi' - 'pdco' - 'qpng'
> You can solve MIQP problems using: 'gurobi'
> You can solve NLP problems using: 'matlab' - 'pdco'

> Checking for available updates ...  
> There are 3494 new commit(s) on <master> and 0 new commit(s) on <develop> [5a1c95 @ master]
> You can update the COBRA Toolbox by running updateCobraToolbox() (from within MATLAB).

Select the optimizer solver.

```matlab
% changeCobraSolver('glpk','all');
changeCobraSolver('gurobi17','all');
```

```matlab
> CBT_LP_SOLVER has been set to gurobi.
> CBT_MILP_SOLVER has been set to gurobi.
> CBT_QP_SOLVER has been set to gurobi.
> CBT_MIQP_SOLVER has been set to gurobi.
> CBT_NLP_SOLVER has been set to gurobi.
```

```matlab
% changeCobraSolver('tomlab_cplex','all');
% changeCobraSolver('gurobi6','all');
```

Load the E.coli iJO1366 model.

```matlab
load('iJO1366.mat');
saved_model = iJO1366;
model = saved_model;
```

**PROCEDURE**

1. **Valine, Leucine and Isoleucine Metabolism**

   The purpose of this tutorial is to identify and review the structure and capabilities of the "Valine, Leucine, and Isoleucine Metabolism" subsystem of the *E.coli* iJO1366 model. This will begin with
an overview of the complete subsystem. This overview will be followed by more detailed descriptions of the individual L-Valine, L-Leucine and L-Isoleucine biosynthesis pathways. It will conclude with a simulation that shows the maximum flux that each of these amino acids can produce in an oxidative range from anaerobic to aerobic conditions.

2. "Valine, Leucine, and Isoleucine Metabolism" Subsystem

The reactions associated with the "Valine, Leucine, and Isoleucine Metabolism" subsystem can be extracted from the model as shown below.

```plaintext
model = saved_model;
model = changeRxnBounds(model,'EX_glc__D_e',-10,1); % Set maximum glucose uptake
model = changeRxnBounds(model,'EX_o2_e',-30,1); % Set oxygen uptake
model = changeRxnBounds(model,'BIOMASS_Ec_iJO1366_WT_53p95M',-0.1); % Disable WT biomass reaction
model = changeObjective(model,'BIOMASS_Ec_iJO1366_core_53p95M'); % Set the objective function
ValineLeucineIsoleucineSubSystems = {'Valine, Leucine, and Isoleucine Metabolism'};
ValineLeucineIsoleucineReactions = model.rxns(ismember(model.subSystems,ValineLeucineIsoleucineSubSystems));
[tmp,ValineLeucineIsoleucine_rxnID] = ismember(ValineLeucineIsoleucineReactions,model.rxns);
reactionNames = model.rxnNames(ValineLeucineIsoleucine_rxnID);
reactionFormulas = printRxnFormula(model,ValineLeucineIsoleucineReactions,0);
T = table(reactionNames,reactionFormulas,'RowNames',ValineLeucineIsoleucineReactions)
fid = 1;
fprintf(fid,'%s %-12s %-70s %-50s\n\n','Reaction','Reaction Name','Reaction Formula');
```

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Reaction Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACHBS</td>
<td>2-aceto-2-hydroxybutanoate synthase</td>
</tr>
<tr>
<td>ACLS</td>
<td>Acetolactate synthase</td>
</tr>
<tr>
<td>DHAD1</td>
<td>Dihydroxy-acid dehydratase (2,3-dihydroxy-3-methylbutanoate)</td>
</tr>
<tr>
<td>DHAD2</td>
<td>Dihydroxy-acid dehydratase (2,3-dihydroxy-3-methylpentanoate)</td>
</tr>
<tr>
<td>ILETA</td>
<td>Isoleucine transaminase</td>
</tr>
<tr>
<td>IPMD</td>
<td>3-isopropylmalate dehydrogenase</td>
</tr>
<tr>
<td>IPPM1a</td>
<td>3-isopropylmalate dehydratase</td>
</tr>
<tr>
<td>IPPM1b</td>
<td>2-isopropylmalate hydratase</td>
</tr>
<tr>
<td>IPPS</td>
<td>2-isopropylmalate synthase</td>
</tr>
<tr>
<td>KARA1</td>
<td>Ketol-acid reductoisomerase (2,3-dihydroxy-3-methylbutanoate)</td>
</tr>
<tr>
<td>KARA2</td>
<td>Ketol-acid reductoisomerase (2-Acetolactate)</td>
</tr>
<tr>
<td>LEUTa</td>
<td>Leucine transaminase (irreversible)</td>
</tr>
<tr>
<td>OMDC</td>
<td>2-Oxo-4-methyl-3-carboxypentanoate decarboxylation</td>
</tr>
<tr>
<td>THIRD_L</td>
<td>L-threonine deaminase</td>
</tr>
<tr>
<td>VALTA</td>
<td>Valine transaminase</td>
</tr>
<tr>
<td>VPAMTr</td>
<td>Valine-pyruvate aminotransferase</td>
</tr>
</tbody>
</table>

2but_c + h_c + pyr_c -> 2ahbut_c + co2_c
h_c + 2 pyr_c -> alac__5_c + co2_c
23dhmb_c -> 3mob_c + h2o_c
23dhnp_c -> 3mob_c + h2o_c
akg_c + ile__6_c -> 3mob_c + glu__4_c
3c2hmp_c + nad_c -> 3c4hmp_c + h_c + nadh_c
2lppm_c + h2o_c -> 3chmp_c
3mob_c + accao_c + h2o_c -> 3c3hmp_c + coa_c + h_c
2ahbut_c + h_c + nadp_c -> 23dhmp_c + nadp_c
4mob_c + glu__4_c -> akg_c + leu__6_c
3c4hmp_c + h_c -> 4mob_c + co2_c
thr__6_c -> 2obut_c + nh4_c
akg_c + val__6_c -> 3mob_c + glu__4_c
3mob_c + ala__6_c -> pyr_c + val__6_c

Table 1. Reaction names and formulas for the "Valine, Leucine, and Isoleucine Metabolism" subsystem.

The connectivity between these reactions can be visualized through an Escher [3] map of this "Valine, Leucine, and Isoleucine Metabolism" subsystem as shown below (Valine_Leucine_Isoleucine_Metabolism_Subsystem.json).
The "Valine, Leucine, and Isoleucine Metabolism" contains pathways for the biosynthesis and production of L-Valine, L-Leucine, and L-Isoleucine. The shortest of these pathways is for the production of L-Valine. There are five reactions involved in the primary production pathway of L-Valine, 'ACLS', 'KARA1', 'DHAD1', 'VALTA' and 'VPAMTr'. The pathways begins with the precursor Pyruvate, pass through three reaction ('ACLS', 'KARA1', 'DHAD1') and then obtains its amino group from ether L-Glutamate (VALTA) in an aerobic environment (Valine_Leucine_Isoleucine_MetabolismSubsystem_Aerobic.png) or L-Alanine (VPAMTr) under anaerobic conditions (Valine_Leucine_Isoleucine_MetabolismSubsystemAnaerobic.png).

The biosynthesis pathway for L-Leucine involves the first three reactions ('ACLS', 'KARA1', 'DHAD1') required for the L-Valine production in addition to six other reactions, including 'IPPS', 'IPPMib', 'IPPMia', 'IPMD', 'OMCDC' and 'LEUTA'. This pathway requires the precursors Pyruvate and Acetyl-CoA in addition to the amino acid L-Glutamate which provides L-Leucine's amino group.

The biosynthesis pathway for L-Isoleucine begins with the amino acid L-Threonine and then followed with five reactions, including 'THRD', 'ACHBS', 'KARA2', 'DHAD2', and 'ILETA'. This pathway also requires the precursor Pyruvate in addition to the amino acid L-Glutamate to provide L-Isoleucine's amino group.

Now let's begin exploring the biosynthesis pathways for L-Valine, L-Leucine, and L-Isoleucine.

3. L-Valine Biosynthesis

The chemical structure for L-Valine (C₆H₁₁NO₂) is shown below.

![L-Valine Chemical Structure](L-Valine.png)
L-Valine is a hydrophobic amino acid.

3.1. L-Valine Biosynthesis Pathways

According to the "Valine and Leucine Metabolism" subsystem there are five reactions involved in the primary production pathway of L-Valine: 'ACLS', 'KARA1', 'DHAD1', 'VALTA' and 'VPAMTr'. The pathways begins with the precursor Pyruvate, pass through three reaction ('ACLS', 'KARA1', 'DHAD1') and then obtains its amino group from ether L-Glutamate (VALTA) or L-Alanine (VPAMTr). To explore the possibility of other L-Valine production pathways we can use the surfNet function we can identify all the reactions that can directly produce L-Valine.
Table 2. Consuming and producing reactions for L-Valine

By looking at the bottom of this print-out we can see that there are three reactions, when operating in the forward direction, can produce L-Valine; including 'VALabcpp', 'VALI2rpp', 'VPAMTr'. Notice that two of these reactions, 'VALI2rpp' and 'VPAMTr', are reversible which implies that flux can flow in both directions allowing them to be both producers of L-Valine and consumers. By looking at the "Consuming reactions" listed about the "Producing reactions, 'VALA', is also reversible allowing it to also be L-Valine producers.

We using the Escher tool [3] can now build a representation of all the reactions and pathways that can lead to the biosynthesis of L-Valine.
From this figure, it can be seen that there are only two major pathways that can produce L-Valine. The first is the primary pathway (‘ACLS’, ‘KARA1’, ‘DHAD1’, ‘VALTA’ and ‘VPAMTr’) and the second is the transporting of L-Valine from the extracellular space (‘EX_val_L-e’; ‘VALtex’, ‘VALabcpp’, and ‘VALt2rpp’).

A table showing these reactions, the reaction name and the reaction formula is given below.

```plaintext
ValineReactions = transposed({'ACLS', 'KARA1', 'DHAD1', 'VALTA', 'VPAMTr', ...
   'EX_val_L-e', 'VALtex', 'VALabcpp', 'VALt2rpp'});
[tmp,Valine_rxnID] = ismember(ValineReactions,model.rxns);
reactionNames = model.rxnNames(Valine_rxnID);
reactionFormulas = printRxnFormula(model,ValineReactions,0);
reactionSubsystem = model.subSystems(Valine_rxnID);
% T = table(reactionNames,reactionSubsystem,'RowNames',ValineReactions)
fid = 1;
fprintf(fid,'%18s %68s %50s\n', 'Reaction', 'Reaction Name', 'Reaction Formula');
[nrows,ncols] = size(ValineReactions);
for row = 1:nrows
    fprintf(fid,'%18s %68s %50s\n', ValineReactions(row,:), reactionNames(row,:), reactionFormulas(row,:));
end
```

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Reaction Name</th>
<th>Reaction Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACLS</td>
<td>Acetolactate synthase</td>
<td>h_c + 2 pyr_c -&gt; alac__s_c + co2_c</td>
</tr>
<tr>
<td>KARA1</td>
<td>Ketol-acid reductoisomerase (2,3-dihydroxy-3-methylbutanoate)</td>
<td>23dhmb_c + nadp_c &lt;=&gt; alac__s_c + h_c + nadph_c</td>
</tr>
<tr>
<td>DHAD1</td>
<td>Dihydroxy-acid dehydratase (2,3-dihydroxy-3-methylbutanoate)</td>
<td>23dmb_c -&gt; 3mob_c + h2o_c</td>
</tr>
<tr>
<td>VALTA</td>
<td>Valine transaminase</td>
<td>akg_c + val__l_c &lt;=&gt; 3mob_c + glu__l_c</td>
</tr>
<tr>
<td>VPAMTr</td>
<td>Valine-pyruvate aminotransferase</td>
<td>3mob_c + ala__l_c &lt;=&gt; pyr_c + val__l_c</td>
</tr>
<tr>
<td>EX_val_L-e</td>
<td>L-Valine exchange</td>
<td>val__l_e -&gt;</td>
</tr>
<tr>
<td></td>
<td>L-valine transport via diffusion (extracellular to periplasm)</td>
<td>val__l_e &lt;=&gt; val__l_p</td>
</tr>
<tr>
<td>VALabcpp</td>
<td>L-valine transport via ABC system (periplasm)</td>
<td>atp_c + h2o_c + val__l_p -&gt; adp_c + h_c + pl_c + val__l_c</td>
</tr>
<tr>
<td>VALt2rpp</td>
<td>L-valine reversible transport via proton symport (periplasm)</td>
<td>h_p + val__l_p &lt;=&gt; h_c + val__l_c</td>
</tr>
</tbody>
</table>

Table 3. Reactions names and formulas for L-Valine biosynthesis reactions.

We can also find the subsystems that are associated with each of these reactions.

```plaintext
reactionSubsystems = model.subSystems(Valine_rxnID);
% T = table(reactionNames,reactionSubsystems,'RowNames',ValineReactions)
fid = 1;
fprintf(fid,'%18s %75s %50s\n', 'Reaction', 'Reaction Name', 'Reaction Subsystem');
[nrows,ncols] = size(ValineReactions);
for row = 1:nrows
    fprintf(fid,'%18s %75s %50s\n', ValineReactions(row,:), reactionNames(row,:), reactionSubsystems(row,:));
end
```

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Reaction Name</th>
<th>Reaction Subsystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACLS</td>
<td>Acetolactate synthase</td>
<td>Valine, Leucine, and Isoleucine Metabolism</td>
</tr>
<tr>
<td>KARA1</td>
<td>Ketol-acid reductoisomerase (2,3-dihydroxy-3-methylbutanoate)</td>
<td>Valine, Leucine, and Isoleucine Metabolism</td>
</tr>
<tr>
<td>DHAD1</td>
<td>Dihydroxy-acid dehydratase (2,3-dihydroxy-3-methylbutanoate)</td>
<td>Valine, Leucine, and Isoleucine Metabolism</td>
</tr>
<tr>
<td>VALTA</td>
<td>Valine transaminase</td>
<td>Valine, Leucine, and Isoleucine Metabolism</td>
</tr>
<tr>
<td>VPAMTr</td>
<td>Valine-pyruvate aminotransferase</td>
<td>Valine, Leucine, and Isoleucine Metabolism</td>
</tr>
</tbody>
</table>
### Table 4. Reactions names and subsystems for L-Valine biosynthesis reactions.

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Description</th>
<th>Subsystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>EX_val_l_e</td>
<td>L-Valine exchange</td>
<td>Extracellular exchange</td>
</tr>
<tr>
<td>VAL tex</td>
<td>L-Valine transport via diffusion (extracellular to periplasm)</td>
<td>Transport, Outer Membrane Porin</td>
</tr>
<tr>
<td>VALabcpp</td>
<td>L-Valine transport via ABC system (periplasm)</td>
<td>Transport, Inner Membrane</td>
</tr>
<tr>
<td>VALt2rpp</td>
<td>L-Valine reversible transport via proton symport (periplasm)</td>
<td>Transport, Inner Membrane</td>
</tr>
</tbody>
</table>

### 3.2 L-Valine Aerobic and Anaerobic Operation

Now let's explore the flux through these L-Valine pathways under normal aerobic conditions.

```python
model = saved_model;
model = changeRxnBounds(model,'EX_glc_D_e',-10,1); % Set maximum glucose uptake
model = changeRxnBounds(model,'EX_o2_e',-30,1); % Set oxygen uptake
model = changeRxnBounds(model,'BIOMASS_Fc_iJO1366_WT_53p9SM',0,'b'); % Disable WT biomass reaction
model = changeObjective(model,'BIOMASS_Fc_iJO1366_core_53p9SM'); % Set the objective function
FBA_solution = optimizeBModel(model,'max',0,0); % Perform FBA
printLabeledData(ValineReactions, round(FBA_solution.x(Valine_rxnID),3))
```

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACLS</td>
<td>0.859</td>
</tr>
<tr>
<td>KARA1</td>
<td>-0.859</td>
</tr>
<tr>
<td>DHAD1</td>
<td>0.859</td>
</tr>
<tr>
<td>VALTA</td>
<td>-0.416</td>
</tr>
<tr>
<td>VPAMTr</td>
<td>0</td>
</tr>
<tr>
<td>EX_val_l_e</td>
<td>0</td>
</tr>
<tr>
<td>VAL tex</td>
<td>0</td>
</tr>
<tr>
<td>VALabcpp</td>
<td>0</td>
</tr>
<tr>
<td>VALt2rpp</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 5. L-Valine fluxes under aerobic conditions.

This can be seen using an Escher [3] plot with the "Aerobic_Reaction Flux.csv" data.
Figure 4. L-Valine production under aerobic conditions (Valine_Biosynthesis_Aerobic.png or Valine_Biosynthesis_Aerobic.svg).

It can be seen that all the L-Valine flux is produced through the primary pathway.

Since there is a difference in the L-Valine pathway under anaerobic conditions lets explore the difference.

```plaintext
model = saved_model;
model = changeRxnBounds(model,'EX_glc__D_e',-10,'l'); % Set maximum glucose uptake
model = changeRxnBounds(model,'EX_o2_e',-10,'l'); % Set oxygen uptake
model = changeRxnBounds(model,'BIOMASS_Ec_iJO1366_WT_53p95M',-10,'b'); % Disable WT biomass reaction
model = changeObjective(model,'BIOMASS_Ec_iJO1366_core_53p95M'); % Set the objective function
FBA_solution = optimizeBModel(model,'max',0,0); % Perform FBA
printLabeledData(ValineReactions, round(FBA_solution.x(Valine_rxnID),3))
```

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Flux (mmol/gDW/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACLS</td>
<td>0.211</td>
</tr>
<tr>
<td>KARA1</td>
<td>-0.211</td>
</tr>
<tr>
<td>DHAD1</td>
<td>0.211</td>
</tr>
<tr>
<td>VALTA</td>
<td>-0.102</td>
</tr>
<tr>
<td>VPAMTr</td>
<td>0</td>
</tr>
<tr>
<td>EX_val__l_e</td>
<td>0</td>
</tr>
<tr>
<td>VALtex</td>
<td>0</td>
</tr>
<tr>
<td>VALabcpp</td>
<td>0</td>
</tr>
<tr>
<td>VALt2rpp</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5. L-Valine fluxes under anaerobic conditions.
This can be seen using an Escher [3] plot with the "Anaerobic_Reaction_Flux.csv" data.

Figure 5. L-Valine production under anaerobic conditions (Valine_Biosynthesis_Anaerobic.png or Valine_Biosynthesis_Anaerobic.svg).

Note that the difference in aerobic vs. anaerobic is that the reaction 'VPAMTr' replaces the reaction 'VALTA'.

3.3 Excess L-Valine Production

When a cell is producing a recombinant protein, it might be required to produce additional L-Valine for the desired bioprocess. What is the maximum amount of excess L-Valine, beyond what the cell needs for normal growth, that can be produced for a given growth-rate?

```matlab
model = saved_model;
model = changeRxnBounds(model, 'EX_glc__D_e', -10, 'l'); % Set maximum glucose uptake
model = changeRxnBounds(model, 'EX_o2_e', 30, 'l'); % Set oxygen uptake
model = changeRxnBounds(model, 'BIOMASS_Ec_iJO1366_WT_53p95M', -3, 'l'); % Disable WT biomass reaction
model = changeObjective(model, 'BIOMASS_Ec_iJO1366_core_53p95M'); % Set the objective function
FBA_solution = optimizeCModel(model, 'max'); % Perform FBA to find optimal growth-rate
model = changeRxnBounds(model, 'BIOMASS_Ec_iJO1366_WT_53p95M', FBA_solution.f, 'l'); % Set fixed growth-rate
model = addDemandReaction(model, 'val__L_c');

DM_val__L_c val__L_c ->
[tmp,Valine_MAX_rxnID] = ismember({'DM_val__L_c'}, model.rxns);
```
model = changeObjective(model,'DM_val__L-c'); % Set the objective function
FBASolution.Valine = optimizeCbModel(model,'max'); % Perform FBA to find optimal growth-rate
xMin = 0.5;
xMax = FBASolution.f;
xInc = (xMax - xMin)/20;
x = xMin;
excessValine = [];
growthRate = [];
for i = 1:21
    model = changeRxnBounds(model,'BIOMASS_Fc_IJ01366_WT_53p95M',x,'b'); % Set fixed growth-rate
    FBASolution.Valine = optimizeCbModel(model,'max'); % Perform FBA
    excessValine(i) = FBASolution.Valine.f;
growthRate(i) = x;
x = x + xInc;
end
plot(growthRate,excessValine)
title('Excess L-Valine');
xlabel('Growth-rate (h-1)'); ylabel('Excess L-Valine (mmol.gDW-1.h-1)');

![Excess L-Valine](image)

Figure 6. A plot showing the maximum amount of L-Valine that can be produced for a given growth-rate.

This figure illustrates that as the need for excess L-Valine increases the growth-rate will need to decrease. In this figure we can see that the excess L-Valine flux can increase from 0.04372 mmol gDW^{-1} hr^{-1} when the cell is at maximum growth-rate to 5.109 mmol gDW^{-1} hr^{-1} when it is at 50% of that optimal growth-rate. Finally, to increase the L-Valine flux beyond these levels will require using the pathways that allow for the transport of L-Valine from the extracellular media.

4. L-Leucine Biosynthesis

The chemical structure for L-Leucine (C₅H₁₁NO₂) is shown below.
4.1. L-Leucine Biosynthetic Pathways

According to the "Valine and Leucine Metabolism" subsystem in Figure 1 there are only pathway that can produce cytoplasmic L-Leucine. The pathway begins with the precursor Pyruvate and includes the following reactions: 'ACLS,' 'KARA1,' 'DHAD1,' 'IPPS,' 'IPPMb,' 'IPPMla,' 'IPMD,' 'OMDC' and 'LEUTA'.

To explore the possibility of other L-Leucine production pathways, we can use the "surfNet" COBRA toolbox function to identify all the reactions that can produce L-Leucine.
+ 0.06715  fe2_c + 0.08700  fe3_c + 0.26316  gln__l_c + 0.26316  glu__l_c
+ 0.61264  gly_c + 0.2151  gtp_c + 48.6015  h2o_c + 0.094738  hls__l_c
+ 0.29068  ile__l_c + 0.19519  k_c + 0.45053  leu__l_c + 0.34316  lys__l_c
+ 0.15369  met__l_c + 0.008675  mg2_c + 0.000223  mthf_c + 0.0000691  mn2_c
+ 7e-08  mobd_c + 0.001831  nadc_c + 0.000447  naco_c + 0.013013  nh4_c
+ 0.00032 ni2_c + 0.000368  pe160_c + 0.054154  pe161_c + 0.18527  phe__l_c
+ 0.000023  pheme_c + 0.22106  pro__l_c + 0.000023  pydx5p_c +
+ 0.000023  ribflv_c + 0.21579  ser__l_c + 0.000023  shene_c + 0.004338  sod_c +
+ 0.000023  thf_c + 0.000223  thmp_c + 0.25369  thr__l_c + 0.056843  trp__l_c +
+ 0.1379  tyr__l_c + 5.5e-05  u1cpdp2_c + 0.1441  utp_c + 0.42336  val__l_c +
+ 0.000341  zn2_c + 0.019456  kdo2lip1d4_c + 0.013894  mure15pxa_p + 0.045946  pe160_p
+ 0.02106  pe161_p -> 53.95  adgc_c + 53.95  h_c + 53.9457  plc_c +
+ 0.7739  pf_c

#1625 LEUTRS Leucyl-tRNA synthetase
  atp_c + leu__l_c + trnaleu_c -> amp_c + leutrna_c + pf_c

Producing reactions:
#1624 LEUTAI Leucine transaminase (irreversible)
  amop_c + glu__l_c: -> akg_c + leu__l_c

#1626 LEUabcpp L-leucine transport via ABC system (periplasm)
  atp_c + h2o_c + leu__l_p: -> adp_c + h_c + leu__l_c + pf_c

#1627 LEU12rpp L-leucine reversible transport via proton symport (periplasm)
  h_p + leu__l_p <- h_c + leu__l_c

Show previous steps...

Table 6. Consuming and producing reactions for L-Leucine

By looking at the bottom of this print-out we can see that there are three reactions, when operating in the forward direction, can produce L-Leucine; including 'LEUTAI', 'LEUabcpp', 'LEU12rpp'. Notice that only one of these reactions, 'LEU12rpp', are reversible which implies that flux can flow in both directions allowing them to be both a produces of L-Leucine and a consumer. By looking at the "Consuming reactions" listed above, the "Producing reaction, LEUTRS, is irreversible preventing it from become a L-Leucine producers.

We using the Escher tool [3] can now build a represenation of all the reactions and pathways that can lead to the biosynthesis of L-Leucine.
From this figure it can be seen that there are only two major pathways that can produce L-Leucine. The first is the primary pathway (‘ACLS’, ‘KARA1’, ‘DHAD1’, ‘IPPS’, ‘IPPMib’, ‘IPPMia’, ‘IPMD’, ‘OMCDC’ and ‘LEUTAI’) and the second is the transporting of L-Leucine from the extracellular space (‘EX_leu__L_e’, ‘LEUtext’, ‘LEUabcpp’, and ‘LEUT2rpp’).

A table showing these reactions, the reaction name and the reaction formula is given below.

```
LeucineReactions = transpose([ACLS', 'KARA1', 'DHAD1', 'IPPS', 'IPPMib', 'IPPMia', 'IPMD', 'OMCDC', 'LEUTAI', ... 'EX_leu__L_e', 'LEUtext', 'LEUabcpp', 'LEUT2rpp']);
[tmp,Leucine_rxnID] = ismember(LeucineReactions,model.rxns);
reactionNames = model.rxnNames(Leucine_rxnID);
reactionFormulas = printRxnFormula(model,LeucineReactions,0);
reactionSubsystem = model.subSystems(Leucine_rxnID);

T = table(reactionNames,reactionSubsystem,'RowNames',LeucineReactions);
fid = 1;
fprintf(fid,'%18s %8s %15s\n','Reaction','Reaction Name','Reaction Formula');

[nrows,ncols] = size(LeucineReactions);
for row = 1:nrows
    fprintf(fid,'%s  %s  %s\n',T{row,'Reaction Name'},T{row,'Reaction'},T{row,'Reaction Formula'});
end
```
Table 7. Reactions names and formulas for L-Leucine biosynthesis reactions.

We can also find the subsystems that are associated with each of these reactions.

Table 8. Reactions names and subsystems for L-Leucine biosynthesis reactions.

4.2. L-Leucine Aerobic Operation

Now let’s explore the flux through these L-Leucine pathways under normal aerobic conditions.
model = changeRxnBounds(model,'EX_glc__D_e',-10,'l'); % Set maximum glucose uptake
model = changeRxnBounds(model,'EX_o2_e',-30,'l'); % Set oxygen uptake
model = changeRxnBounds(model,'BIOMASS_Ec_iJO1366_WT_53p95M',-10,'b'); % Disable WT biomass reaction
model = changeObjective(model,'BIOMASS_Ec_iJO1366_core_53p95M'); % Set the objective function
FBAsolution = optimizeBModel(model,'max',0,0); % Perform FBA
printLabeledData(LeucineReactions, round(FBAsolution.x(Leucine_rxnID),3))

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Flux</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACLS</td>
<td>0.859</td>
</tr>
<tr>
<td>KARA1</td>
<td>-0.859</td>
</tr>
<tr>
<td>DHAD1</td>
<td>0.859</td>
</tr>
<tr>
<td>IPPS</td>
<td>0.443</td>
</tr>
<tr>
<td>IPPM1b</td>
<td>-0.443</td>
</tr>
<tr>
<td>IPPM1a</td>
<td>-0.443</td>
</tr>
<tr>
<td>IPMD</td>
<td>0.443</td>
</tr>
<tr>
<td>OMCDC</td>
<td>0.443</td>
</tr>
<tr>
<td>LEUTAI</td>
<td>0.443</td>
</tr>
<tr>
<td>EX_leu__l_e</td>
<td>0</td>
</tr>
<tr>
<td>LEUtxex</td>
<td>0</td>
</tr>
<tr>
<td>LEUbcpp</td>
<td>0</td>
</tr>
<tr>
<td>LEU2rpp</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 9.** L-Leucine fluxes under aerobic conditions.

This can be visualized using an Escher [3] plot with the "Aerobic_Reaction_Flux.csv" data.
4.3 Excess L-Leucine Production

When a cell is producing a recombinant protein, it might be required to produce additional L-Leucine for the desired bioproduct. What is the maximum amount of excess L-Leucine, beyond what the cell needs for normal growth, that can be produced for a given growth-rate?

```python
code_snippet
model = saved_model;
model = changeRxnBounds(model,'EX_glc__D_e',-10,'1'); % Set maximum glucose uptake
model = changeRxnBounds(model,'EX_o2_e',-30,'1'); % Set oxygen uptake
model = changeRxnBounds(model,'BIOMASS_Ec_iJO1366_WT_53p95M','-0','b'); % Disable WT biomass reaction
model = changeObjective(model,'BIOMASS_Ec_iJO1366_core_53p95M'); % Set the objective function
FBA solution = optimizeCbModel(model,'max'); % Perform FBA to find optimal growth-rate
model = changeRxnBounds(model,'BIOMASS_Ec_iJO1366_WT_53p95M',FBA solution.f,'b'); % Set fixed growth-rate
model = addDemandReaction(model, 'leu_l_c');

DM_leu_l_c leu_l_c ->
[tmp,Valine_MAX_rxnID] = ismember({'DM_leu_l_c'},model.rxns);
model = changeObjective(model,'DM_leu_l_c'); % Set the objective function
```
FBA_solution.Leucline = optimizeCbModel(model,'max'); % Perform FBA to find optimal growth-rate
xMin = 0.5;
xMax = FBA_solution.f;
xInc = (xMax - xMin)/20;
x = xMin;
excessleucine = [];
growthRate = [];
for i = 1:21
    model = changeRxnBounds(model,'BIOMASS_Ec_iJO1366_WT_53p95M',x,'b'); % Set fixed growth-rate
    FBA_solution.Leucline = optimizeCbModel(model,'max'); % Perform FBA
    excessleucine(i) = FBA_solution.Leucline.f;
growthRate(i) = x;
x = x + xInc;
end
plot(growthRate,excessleucine)
title('Excess L-Leucine');
xlabel('Growth-rate (h⁻¹)'); ylabel('Excess L-Leucine (mmol.gDW⁻¹.h⁻¹)');

![Graph showing the maximum amount of L-Leucine that can be produced for a given growth-rate.](image)

**Figure 10.** A plot showing the maximum amount of L-Leucine that can be produced for a given growth-rate.

This figure illustrates that as the need for excess L-Leucine increases the growth-rate will need to decrease. In this figure we can see that the excess L-Leucine flux can increase from 0.0349 mmol·gDW⁻¹·hr⁻¹ when the cell is at maximum growth-rate to 4.003 mmol·gDW⁻¹·hr⁻¹ when it is at 50% of that optimal growth-rate. Finally, to increase the L-Leucine flux beyond these levels will require using the pathways that allow for the transport of L-Leucine from the extracellular media.

4. L-Isoleucine Biosynthesis

The chemical structure for L-Isoleucine (C₆H₁₃NO₃) is shown below.
4.1. L-Isoleucine Biosynthesis Pathways

According to the “Valine and Leucine Metabolism” subsystem in Figure 1 there are only pathway that can produce cytoplasmic L-Isoleucine. The pathway begins with the amino acid L-Threonine and includes the following reactions: ‘THRD_L’, ‘ACHBS’, ‘KARA2’, ‘DHAD2’, and ‘ILEA’.

To explore the possibility of other L-Valine production pathways we can use the "suffNet" function to identify all the reactions that can produce L-Isoleucine.

```
Met #621 i1le_L_c, L-Isoleucine, C6H13NO2
Consuming reactions: #7 BIOMASS_Ec_iJO1366_WT_53p95M E. coli biomass objective function (iJO1366) - WT - with 53.95 GAM estimate
0.000223 108thf_c + 0.000223 2demol _c + 2.5e-05 2fe2s_c + 0.000248 4fe4s_c + 0.000223 Snthf_c + 0.000279 accoa_c + 0.000223 adocbl_c
+ 0.49915 ala_L_c + 0.000223 anet_c + 0.28742 arg_L_c + 0.23423 asn_L_c
+ 0.23423 asp_L_c + 54.12 atp_c + 0.000116 bncogdp_c + 2e-06 btn_c
+ 0.004952 cas_c + 0.000223 chor_c + 0.000022 c1_c + 0.000168 coa_c
+ 2.4e-05 coxal2_c + 0.1298 ctp_c + 0.000674 cu2_c + 0.088988 cys_L_c
+ 0.024805 datp_c + 0.025612 dctp_c + 0.025612 dgtcp_c + 0.024805 dttcp_c
+ 0.000233 enter_c + 0.000223 rad_c + 0.006388 fe2c_c + 0.007428 fe3_c
+ 0.25571 gln_L_c + 0.25571 glu_L_c + 0.5953 glyc_c + 0.15419 glycogen_c
+ 0.000233 gltadm_c + 0.20912 gtp_c + 0.48.7529 h2o_c + 0.000223 hemc_c
+ 0.009206 his_L_c + 0.28231 i1le_L_c + 0.000065 k_c + 0.03778 leu_L_c
+ 3e-05 lipob_c + 0.33345 lys_L_c + 3.1e-05 malcoa_c + 0.14934 met_L_c
+ 0.008253 mg2_c + 0.000223 mthf_c + 0.000658 mn2_c + 7e-06 mntb_c
+ 7e-06 mncogdp_c + 7e-06 mncogdp_c + 0.000223 mql8_c + 0.001787 nad_c
+ 4.5e-05 nadh_c + 0.001112 nadp_c + 0.000335 nadph_c + 0.001279 nh4_c
+ 0.003067 nh2_c + 0.012366 pe160_c + 0.000618 pe161_c + 0.004957 pe181_c
+ 0.005707 pg160_c + 0.004439 pg161_c + 0.000288 pg181_c + 0.18002 phe_L_c
+ 0.000023 phm_c + 0.2148 pro_L_c + 0.03327 strc_c + 0.000223 pydx5p_c
+ 0.000023 qu82_c + 0.000223 ribflv_c + 0.20686 ser_L_c + 0.000223 sheme_c
+ 0.004126 soud_c + 0.006744 smpd_c + 9.8e-05 succoa_c + 0.000223 tfh_c
+ 0.000023 tnhpp_c + 0.24651 thr_L_c + 0.055234 trp_L_c + 0.13399 tyr_L_c
+ 5.9e-05 utop_c + 0.1401 uro_c + 0.4118 val_L_c + 0.000324 zn2_c
+ 0.008151 cnlp0_p + 0.002944 cnlp10_p + 0.00229 clnlp10_p +
0.00118 clnlp10_p + 0.000135 murein3p3p_p + 0.000005 murein3p3x4p_p +
0.005318 murein4p_p + 0.000548 murein4p4p_p + 0.000673 murein4p5p4p_p +
0.001798 pe160_p + 0.024732 pe161_p + 0.012747 pe181_p + 0.004892 pe180_p +
0.003885 pg161_p + 0.001961 pg181_p + 53.95 adp_c + 53.95 h_c +
53.9495 p_l_c + 0.74683 psl_c
#8 BIOMASS_Ec_iJO1366_core_53p95M E. coli biomass objective function (iJO1366) - core - with 53.95 GAM estimate
0.000223 108thf_c + 2.6e-05 2fe2s_c + 0.000223 2onph_c + 0.00026 4fe4s_c
+ 0.51639 ala_L_c + 0.000223 anet_c + 0.29579 arg_L_c + 0.24105 asn_L_c
+ 0.24105 asp_L_c + 54.12 atp_c + 0.000122 bncogdp_c + 2e-06 btn_c
+ 0.005299 cas_c + 0.005299 cl_c + 0.000576 coa_c + 2.5e-05 coxal2_c
+ 0.13351 ctp_c + 0.000769 cu2_c + 0.09158 cys_L_c + 0.026166 datc_c
+ 0.027017 dgtcp_c + 0.027017 dgtcp_c + 0.026166 dttcp_c + 0.000223 rad_c
+ 0.006715 fe2_c + 0.007888 fe3_c + 0.26316 gln_L_c + 0.26316 glu_L_c
```
From these results we can see that there are two reactions that can directly produce cytoplasmic L-Isoleucine in the forward direction, they include ‘ILEtcp’ and ‘ILEt2rpp’. Notice that only one of these reactions, ‘ILEt2rpp’, are reversible which implies that flux can flow in both directions allowing them to be both a produces of L-isoleucine and a consumer. By looking at the “Consuming reactions” listed above, the ‘Producing reaction, ‘ILETA’, allowing it to be an L-Isoleucine producer.

We using the Escher tool [3] can now build a represenation of all the reactions and pathways that can lead to the biosynthesis of L-Isoleucine.
Figure 12. Biosynthesis pathways for the production of L-Isoleucine (Isoluecine_Biosynthesis.json, Isoleucine_Biosynthesis.png or Isoleucine_Biosynthesis.svg).

From this figure it can be seen that there are only two major pathways that can produce L-Leucine. The first is the primary pathway ('ACLS', 'KARAI', 'DHAD1', 'IPPS', 'IPPMB', 'IPPMia', 'IPMD', 'OMCDC' and 'LEUTAI') and the second is the transporting of L-Leucine from the extracellular space ('EX_leu__L_e', 'LEUTex', 'LEUabcpp', and 'LEUI2ppp').

A table showing these reactions, the reaction name and the reaction formula is given below.

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Reaction Name</th>
<th>Reaction Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACLS</td>
<td>Acetolactate synthase</td>
<td>h_c + 2 pyr_c → alac__S_c + co2_c</td>
</tr>
<tr>
<td>KARAI</td>
<td>Ketol-acid reductoisomerase (2,3-dihydroxy-3-methylbutanoate)</td>
<td>23dhmb_c + nadp_c ↔ alac__S_c + h_c + nadph_c</td>
</tr>
<tr>
<td>DHAD1</td>
<td>Dihydroxy-acid dehydratase (2,3-dihydroxy-3-methylbutanoate)</td>
<td>23dhmb_c → 3mob_c + h2o_c</td>
</tr>
<tr>
<td>IPPS</td>
<td>2-isopropylmalate synthase</td>
<td>3mob_c + ackoa_c + h2o_c → 3chmp__c + coa_c + h_c</td>
</tr>
<tr>
<td>IPPMB</td>
<td>2-isopropylmalate hydratase</td>
<td>2ippm__c + h2o_c ↔ 3c3hmp__c</td>
</tr>
</tbody>
</table>
Table 11. Reactions names and formulas for L-Isoleucine biosynthesis reactions.

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Reaction Name</th>
<th>Reaction Subsystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACLS</td>
<td>Acetolactate synthase</td>
<td>Valine, Leucine, and Isoleucine Metabolism</td>
</tr>
<tr>
<td>KARA1</td>
<td>Ketol-acid reductoisomerase</td>
<td>Valine, Leucine, and Isoleucine Metabolism</td>
</tr>
<tr>
<td>DHAD1</td>
<td>Dihydroxy-acid dehydratase</td>
<td>Valine, Leucine, and Isoleucine Metabolism</td>
</tr>
<tr>
<td>IPPS</td>
<td>2-isopropylmalate synthase</td>
<td>Valine, Leucine, and Isoleucine Metabolism</td>
</tr>
<tr>
<td>IPPM1b</td>
<td>2-isopropylmalate hydratase</td>
<td>Valine, Leucine, and Isoleucine Metabolism</td>
</tr>
<tr>
<td>IPPM1a</td>
<td>3-isopropylmalate dehydratase</td>
<td>Valine, Leucine, and Isoleucine Metabolism</td>
</tr>
<tr>
<td>TPMD</td>
<td>3-isopropylmalate dehydrogenase</td>
<td>Valine, Leucine, and Isoleucine Metabolism</td>
</tr>
<tr>
<td>OMCDC</td>
<td>2-Oxo-4-methyl-3-carboxypentanoate decarboxylation</td>
<td>Valine, Leucine, and Isoleucine Metabolism</td>
</tr>
<tr>
<td>LEUT1</td>
<td>Leucine transaminase (irreversible)</td>
<td>Valine, Leucine, and Isoleucine Metabolism</td>
</tr>
<tr>
<td>EX_leu_l_e</td>
<td>L-Leucine exchange</td>
<td>Extracellular exchange</td>
</tr>
<tr>
<td>LEutex</td>
<td>L-leucine transport via diffusion (extracellular to periplasm)</td>
<td>Transport, Outer Membrane Porin</td>
</tr>
<tr>
<td>LEUabcpp</td>
<td>L-leucine transport via ABC system (periplasm)</td>
<td>Transport, Inner Membrane</td>
</tr>
<tr>
<td>LEUt2rpp</td>
<td>L-leucine reversible transport via proton symport (periplasm)</td>
<td>Transport, Inner Membrane</td>
</tr>
</tbody>
</table>

Table 12. Reactions names and subsystems for L-Isoleucine biosynthesis reactions.

5.2. L-Isoleucine Aerobic Operation

Now let's explore the flux through these L-isoleucine pathways under normal aerobic conditions.

```plaintext
model = saved_model;
IsoleucineReactions = transpose(['ASPTA','ASNS1','ASNS2','EX_asn__L_e','ASPTex','ASPtacb','ASPT2_2pp',
                                 'ASPT2_3pp','ASPT2pp_copy1','ASPT2pp_copy2','SUCASptpp','EX_asn__L_e','ASNTex','ASNpp','ASNacb','
                                 'ASNT2rpp','ASPK']);
[tmp,Isoleucine_rxnID] = ismember(IsoleucineReactions,model.rxns);
model = changeRxnBounds(model,'EX_glc__D_e','-10','1'); % Set maximum glucose uptake
model = changeRxnBounds(model,'EX_o2_e','-30','1'); % Set oxygen uptake
model = changeRxnBounds(model,'BIOMASS_Ec_iJO1366_WT_53p95M','-8','b'); % Disable WT biomass reaction
```
Table 13. L-Isoleucine fluxes values under aerobic conditions.

This can be visualized using an Escher [3] plot with the “Aerobic_Reaction_Flux.csv” data.

Figure 13. Aerobic production of L-Isoleucine (Isoleucine_Biosynthesis_Aerobic.png or Isoleucine_Biosynthesis_Aerobic.svg).
4.3 Excess L-Isoleucine Production

When a cell is producing a recombinant protein, it might be required to produce additional L-Isoleucine for the desired bioprocess. What is the maximum amount of excess L-Isoleucine, beyond what the cell needs for normal growth, that can be produced for a given growth-rate?

```matlab
model = saved_model;
model = changeRxnBounds(model, 'EX_glc_D_e', -10, 'l'); % Set maximum glucose uptake
model = changeRxnBounds(model, 'EX_o2_e', -30, 'l'); % Set oxygen uptake
model = changeRxnBounds(model, 'BIOMASS_Ec_iJO1366_core_53p95M', 0, 'b'); % Disable WT biomass reaction
model = changeObjective(model, 'BIOMASS_Ec_iJO1366_core_53p95M'); % Set the objective function
FBA_solution = optimizeBModel(model, 'max'); % Perform FBA to find optimal growth-rate
model = changeRxnBounds(model, 'BIOMASS_Ec_iJO1366_core_53p95M', FBA_solution.f, 'b'); % Set fixed growth-rate
model = addDemandReaction(model, 'ile__L_c');

DM_ile__L_c   ile__L_c    ->

tmp, Isoleucine_MAX_rxnID = ismember([DM_ile__L_c], model.rxns);
model = changeObjective(model, 'DM_ile__L_c'); % Set the objective function
FBA_solution_Isoleucine = optimizeBModel(model, 'max'); % Perform FBA to find optimal growth-rate
xMin = 0.5;
xMax = FBA_solution.f;
xInc = (xMax - xMin)/20;
x = xMin;
excessIsoleucine = [];
growthRate = [];
for i = 1:21
    model = changeRxnBounds(model, 'BIOMASS_Ec_iJO1366_core_53p95M', x, 'b'); % Set fixed growth-rate
    FBA_solution_Isoleucine = optimizeBModel(model, 'max'); % Perform FBA
    excessIsoleucine(i) = FBA_solution_Isoleucine.f;
    growthRate(i) = x;
    x = x + xInc;
end
plot(growthRate, excessIsoleucine)
title('Excess L-Isoleucine');
xlabel('Growth-rate (h-1)'); ylabel('Excess L-Isoleucine (mmol.gDW-1.h-1)');
```
Figure 14. A plot showing the maximum amount of L-Isoleucine that can be produced for a given growth-rate.

This figure illustrates that as the need for excess L-Isoleucine increases the growth-rate will need to decrease. In this figure we can see that the excess L-Isoleucine flux can increase from 0.03133 mmol·gDW⁻¹·hr⁻¹ when the cell is at maximum growth-rate to 3.567 mmol·gDW⁻¹·hr⁻¹ when it is at 50% of that optimal growth-rate. Finally, to increase the L-Isoleucine flux beyond these levels will require using the pathways that allow for the transport of L-Isoleucine from the extracellular media.

5. Aerobic vs Anaerobic Amino Acid Production

Now let’s look at the total amount of flux that is created for each of these amino acids as the oxygen content varies from anaerobic to aerobic.

```matlab
model = saved_model;
model = changeRxnBounds(model,'EX_glc_D_e','-10','1'); % Set maximum glucose uptake
model = changeRxnBounds(model,'BIOMASS_Ec_iJO1366_WT_S3p95M','-8','9'); % Disable WT biomass reaction
model = changeObjective(model,'BIOMASS_Ec_iJO1366_core_S3p95M'); % Set the objective function
val_flux = [];
leu_flux = [];
ile_flux = [];
for k = 1:31
  model = changeRxnBounds(model,'EX_o2_e',-(k-1),'b'); % Set oxygen uptake
  FBA = optimizeBModel(model,'max',0,0); % Perform FBA
  [P, C, V, P] = computeFluxSplits(model, {'val__l_c'}, FBA.x);
  val_flux(k) = sum(V);
  [P, C, V, P] = computeFluxSplits(model, {'leu__l_c'}, FBA.x);
  leu_flux(k) = sum(V);
  [P, C, V, P] = computeFluxSplits(model, {'ile__l_c'}, FBA.x);
  ile_flux(k) = sum(V);
  growthRate(k) = FBA.x.f;
end
figure(1)
```
Figure 15. A plot showing the total amount of flux that is can be created for each of these amino acids as the oxygen content varies from anaerobic to aerobic.

Note that the noise on the L-Valine chart is the result of loops in the production pathways.

6. Conclusion

The purpose of this tutorial was to identify and review the structure and capabilities of the "Valine and Leucine Metabolism" subsystem of the E.coli iJO1366 model. It began with an overview of the complete subsystem. This was followed by more detailed descriptions of the individual L-Valine, L-Leucine and L-Isoleucine biosynthesis pathways. It concluded with a simulation showing
the maximum flux that each these amino acids can produce in a range from anaerobic to aerobic conditions.

References