COBRA Models

Learning Objectives

Each student should be able to:

- Explain the purpose of the COBRApy Toolbox,
- Understand the principles behind the COBRA models,
- Demonstrate the ability to interact with COBRA models.

COBRA Models Overview

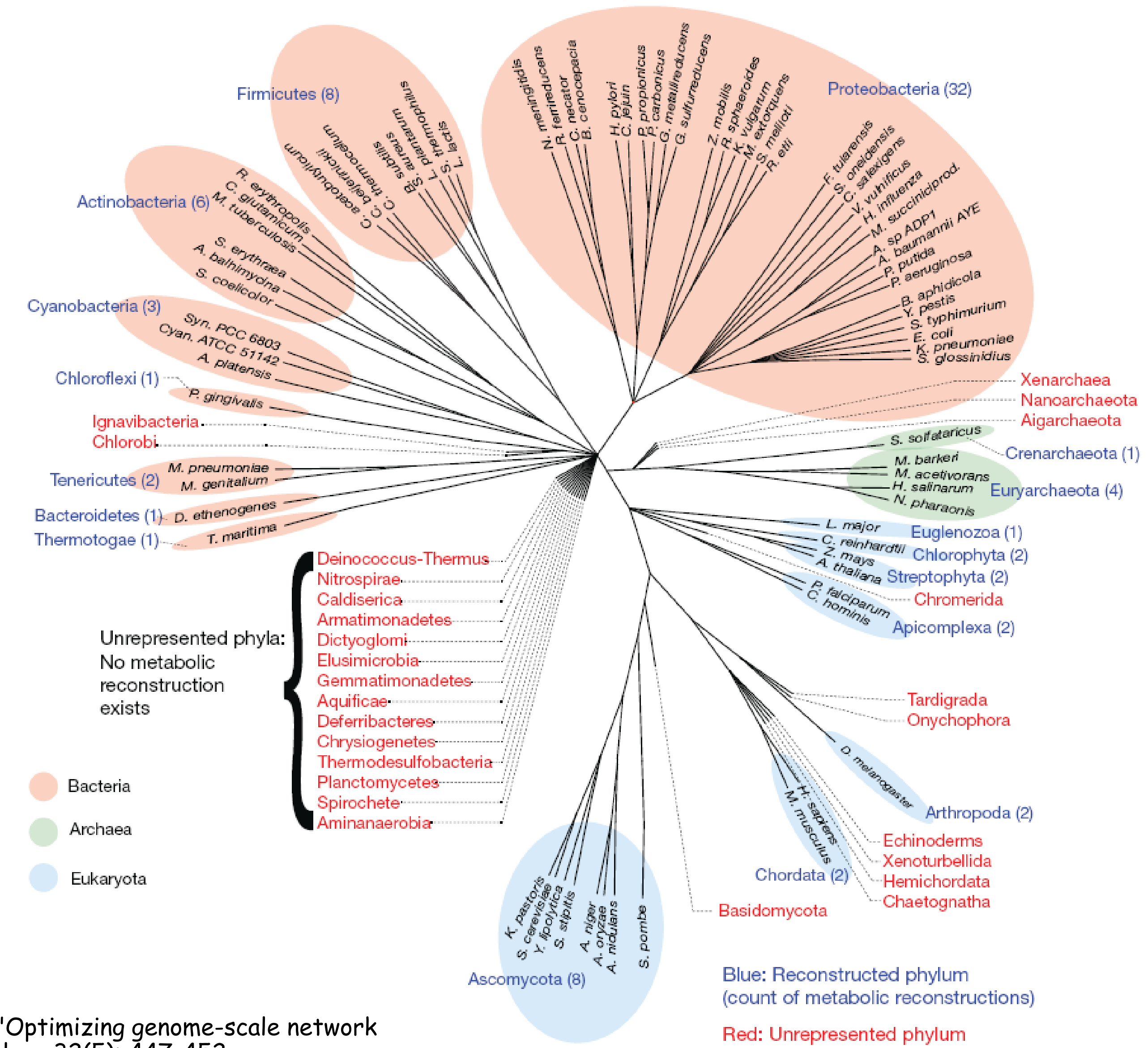
- COBRApy Toolbox Overview
- COBRA Models Overview
  - ✓ COBRApy Examples
- Reactions
  - ✓ COBRApy Examples
- Metabolites
  - ✓ COBRApy Examples
- Genes
  - ✓ COBRApy Examples
- Visualization
  - ✓ COBRApy Examples

The COBRApy Toolbox

Metabolic Reconstructions

A GEnome scale Network Reconstructions (GENREs) serves as a structured knowledge base of established biochemical facts, while a GEnome scale Models (GEMs) is a model which supplements the established biochemical information with additional (potentially hypothetical) information to enable computational simulation and analysis.

Metabolic reconstructions for 78 species across the tree of life

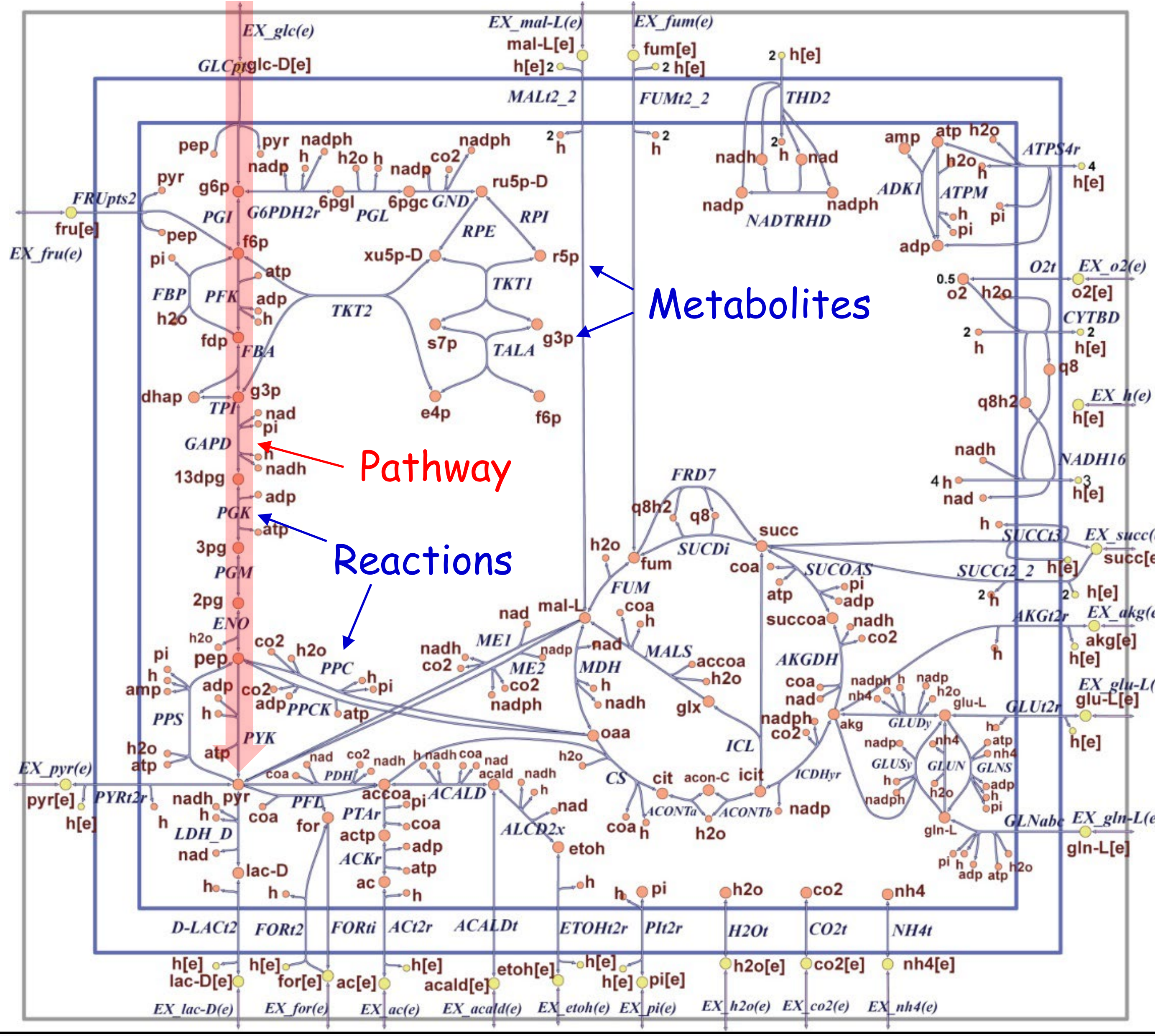


Blue: Reconstructed phylum
(count of metabolic reconstructions)

Red: Unrepresented phylum
(no metabolic reconstructions exist)

Monk, J., J. Nogales, et al. (2014). "Optimizing genome-scale network reconstructions." Nature biotechnology 32(5): 447-452.

Metabolic Models



Metabolites

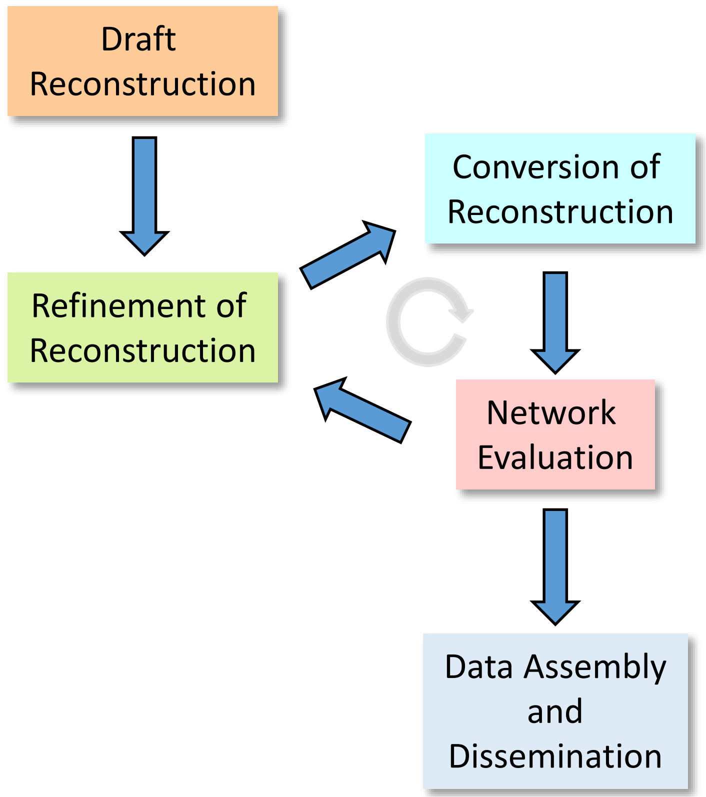
Pathway

Reactions

Orth, J. D., I. Thiele, et al. (2010). "What is flux balance analysis?" Nature biotechnology **28(3): 245-248.**

Genome-scale
Metabolic Reconstruction Process

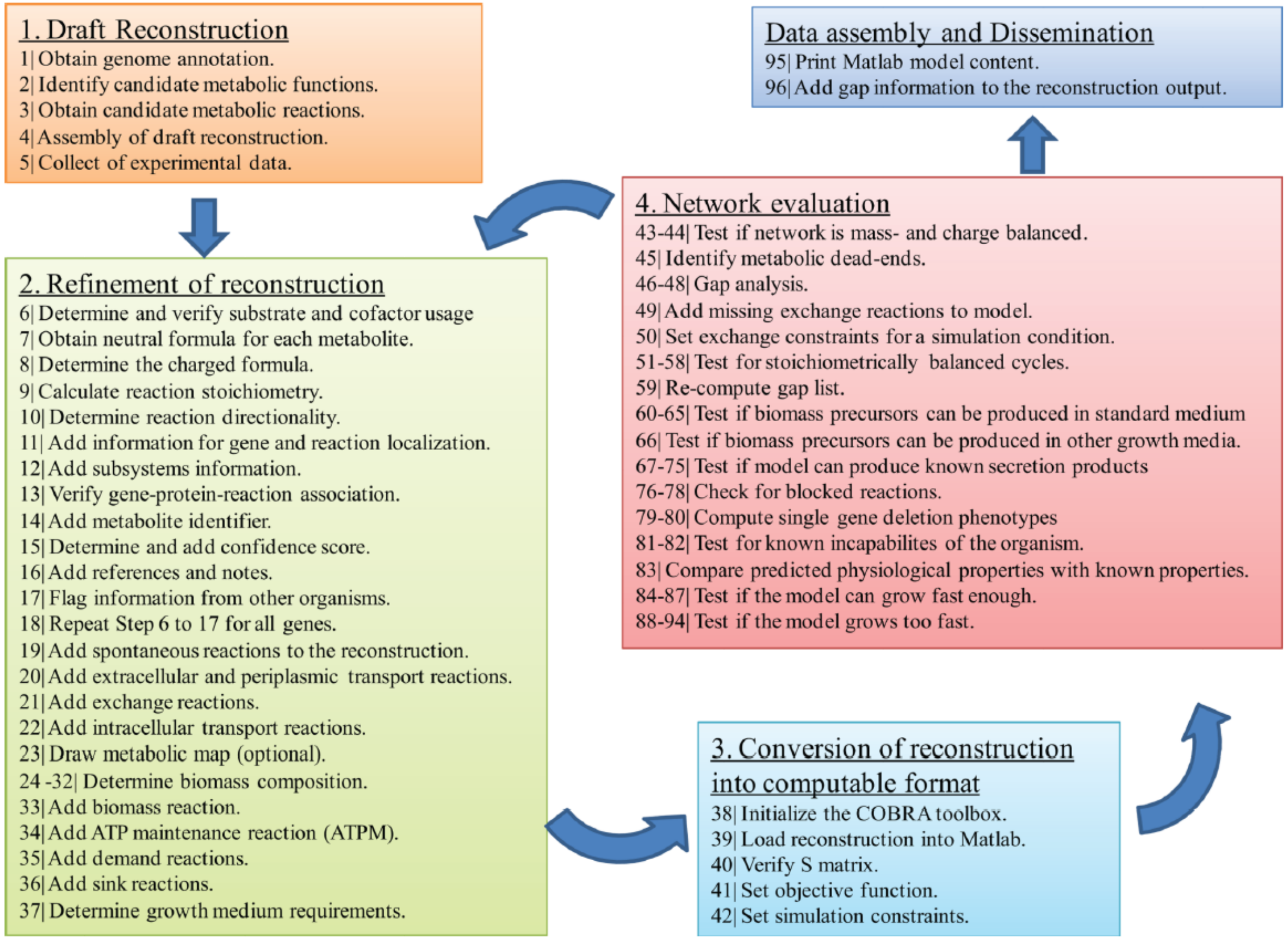
- Draft Reconstruction
- Refinement of Reconstruction
- Conversion of Reconstruction into Computable Format
- Network Evaluation
- Data Assembly and Dissemination


Draft Reconstruction
Refinement of Reconstruction
Conversion of Reconstruction
Network Evaluation
Data Assembly and Dissemination

Thiele, I. and B. O. Palsson (2010). "A protocol for generating a high-quality genome-scale metabolic reconstruction." Nature protocols 5(1): 93-121.

Reconstruction Process: 96 Step Protocol

Thiele, I. and B. O. Palsson (2010). "A protocol for generating a high-quality genome-scale metabolic reconstruction." Nature protocols 5(1): 93-121.



1. Draft Reconstruction

1| Obtain genome annotation.
2| Identify candidate metabolic functions.
3| Obtain candidate metabolic reactions.
4| Assembly of draft reconstruction.
5| Collect of experimental data.

2. Refinement of reconstruction

6| Determine and verify substrate and cofactor usage
7| Obtain neutral formula for each metabolite.
8| Determine the charged formula.
9| Calculate reaction stoichiometry.
10| Determine reaction directionality.
11| Add information for gene and reaction localization.
12| Add subsystems information.
13| Verify gene-protein-reaction association.
14| Add metabolite identifier.
15| Determine and add confidence score.
16| Add references and notes.
17| Flag information from other organisms.
18| Repeat Step 6 to 17 for all genes.
19| Add spontaneous reactions to the reconstruction.
20| Add extracellular and periplasmic transport reactions.
21| Add exchange reactions.
22| Add intracellular transport reactions.
23| Draw metabolic map (optional).
24 -32| Determine biomass composition.
33| Add biomass reaction.
34| Add ATP maintenance reaction (ATPM).
35| Add demand reactions.
36| Add sink reactions.
37| Determine growth medium requirements.

3. Conversion of reconstruction into computable format

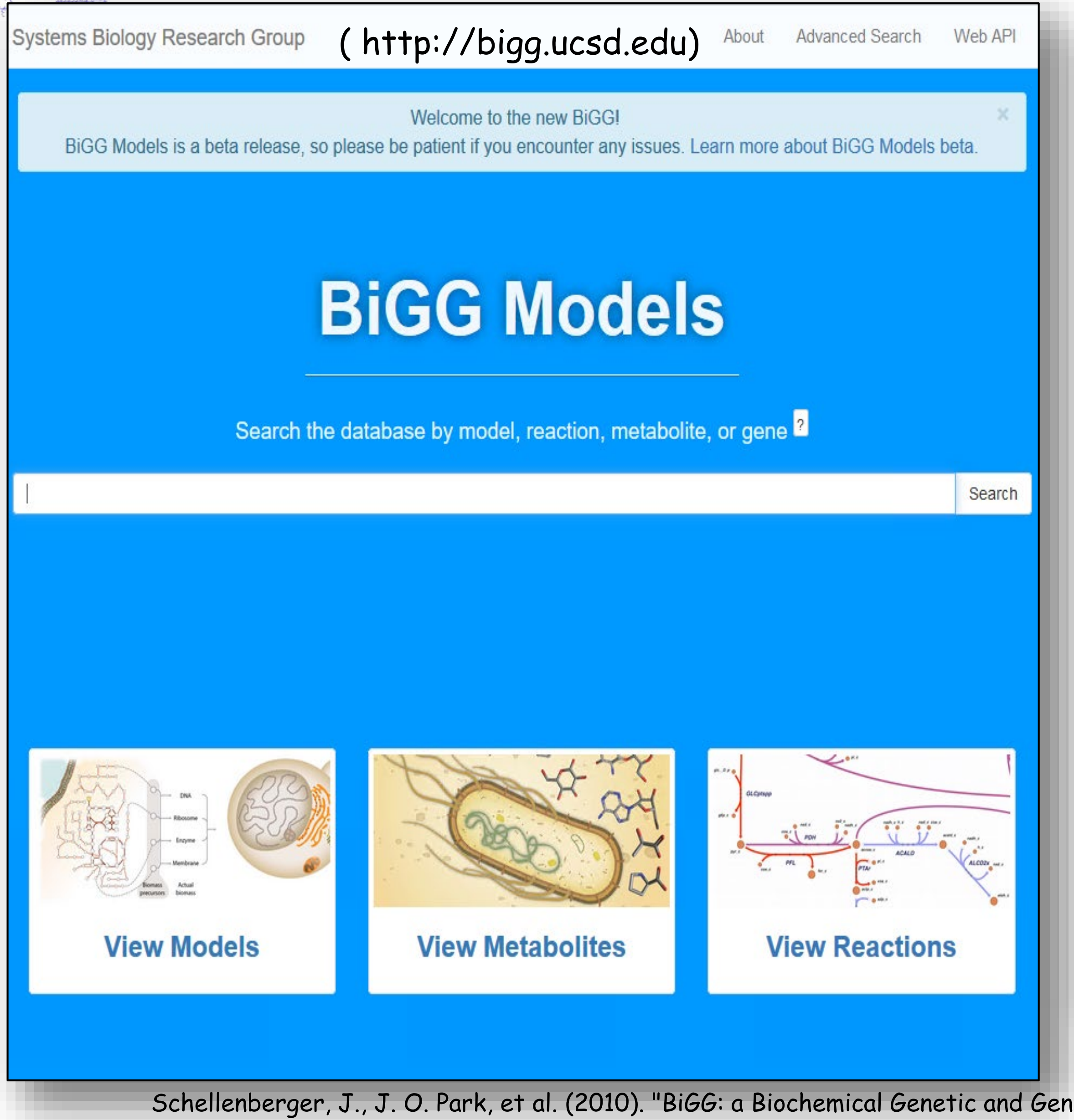
38| Initialize the COBRA toolbox.
39| Load reconstruction into Matlab.
40| Verify S matrix.
41| Set objective function.
42| Set simulation constraints.

4. Network evaluation

43-44| Test if network is mass- and charge balanced.
45| Identify metabolic dead-ends.
46-48| Gap analysis.
49| Add missing exchange reactions to model.
50| Set exchange constraints for a simulation condition.
51-58| Test for stoichiometrically balanced cycles.
59| Re-compute gap list.
60-65| Test if biomass precursors can be produced in standard medium
66| Test if biomass precursors can be produced in other growth media.
67-75| Test if model can produce known secretion products
76-78| Check for blocked reactions.
79-80| Compute single gene deletion phenotypes
81-82| Test for known incapabilites of the organism.
83| Compare predicted physiological properties with known properties.
84-87| Test if the model can grow fast enough.
88-94| Test if the model grows too fast.

Data assembly and Dissemination

95| Print Matlab model content.
96| Add gap information to the reconstruction output.



Systems Biology Research Group (http://bigg.ucsd.edu) About Advanced Search Web API

Welcome to the new BiGG!
BiGG Models is a beta release, so please be patient if you encounter any issues. Learn more about BiGG Models beta.

BiGG Models

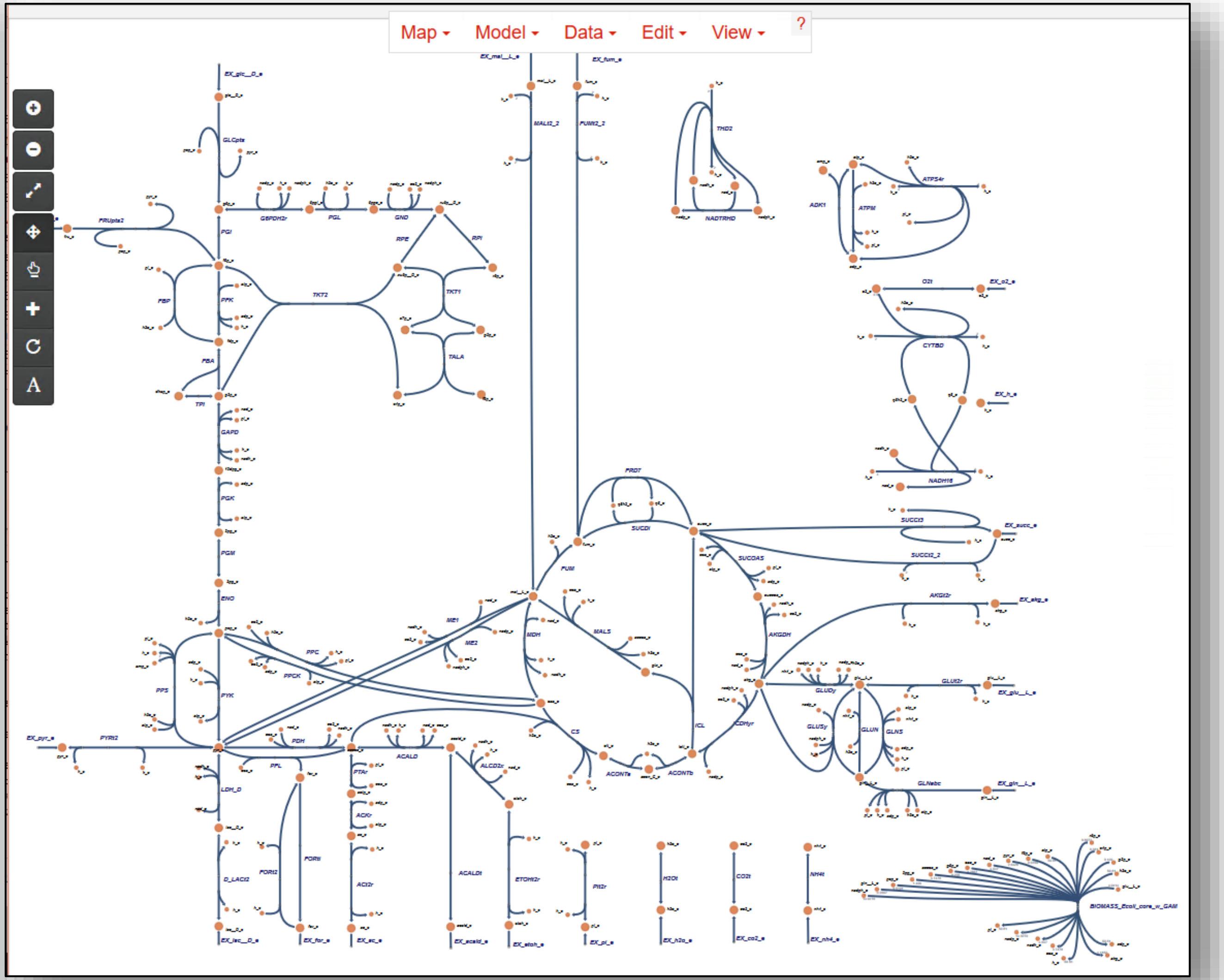
Search the database by model, reaction, metabolite, or gene ?

Search

View Models

View Metabolites

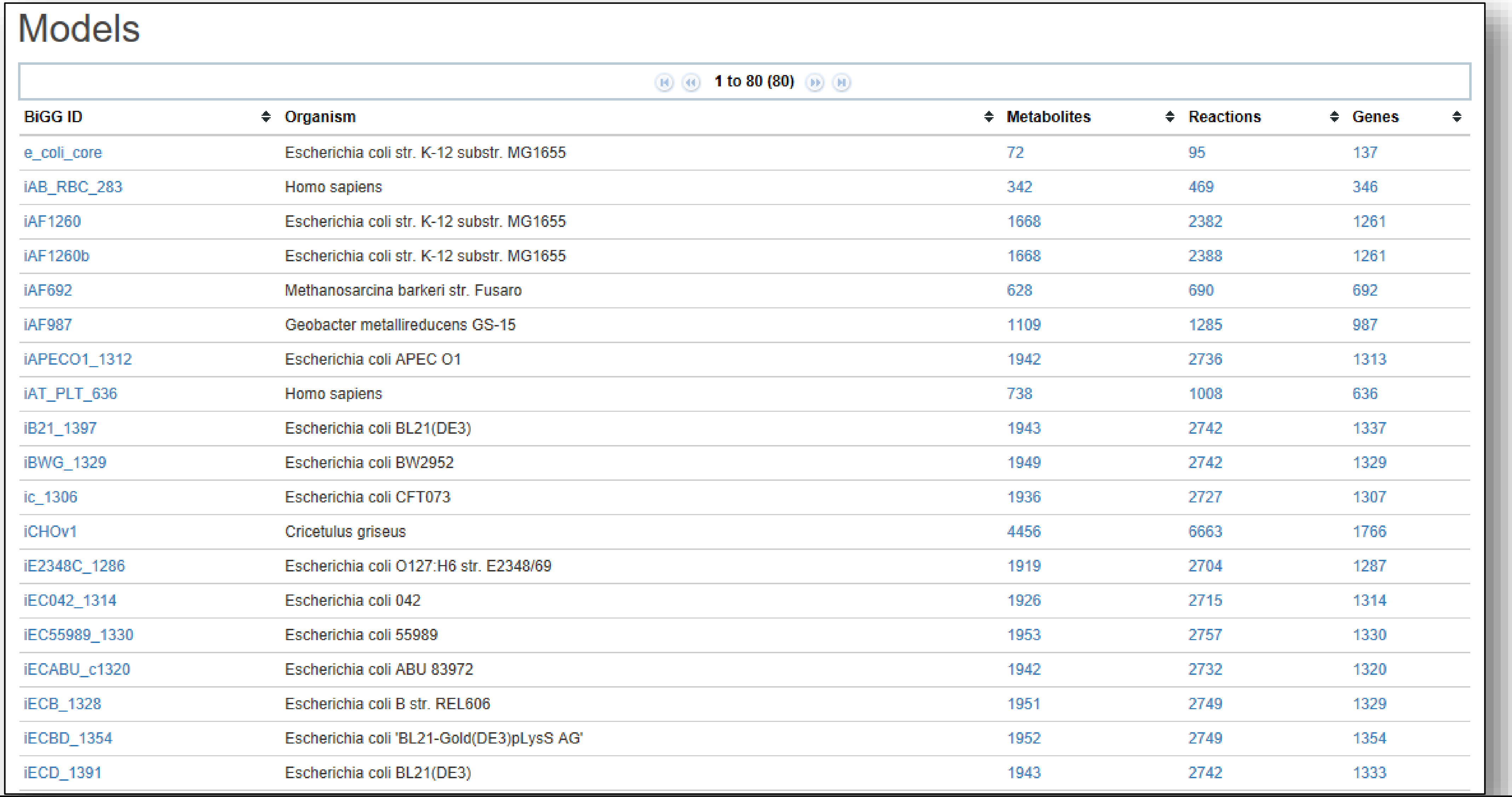
View Reactions



http://bigg.ucsd.edu/models/e_coli_core

Schellenberger, J., J. O. Park, et al. (2010). "BiGG: a Biochemical Genetic and Genomic knowledgebase of large scale metabolic reconstructions." BMC Bioinformatics 11: 213.

BIGG Models at http://bigg.ucsd.edu/models

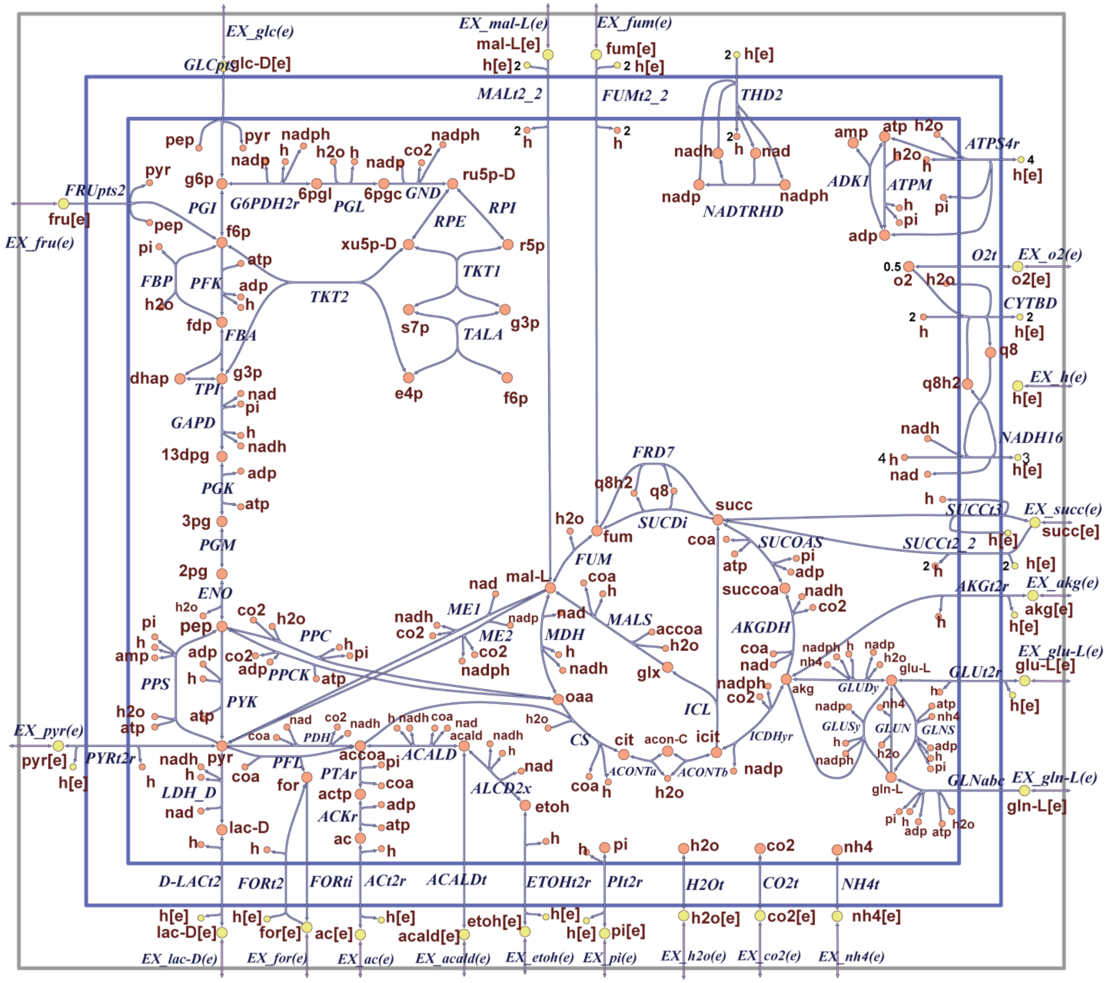


Models

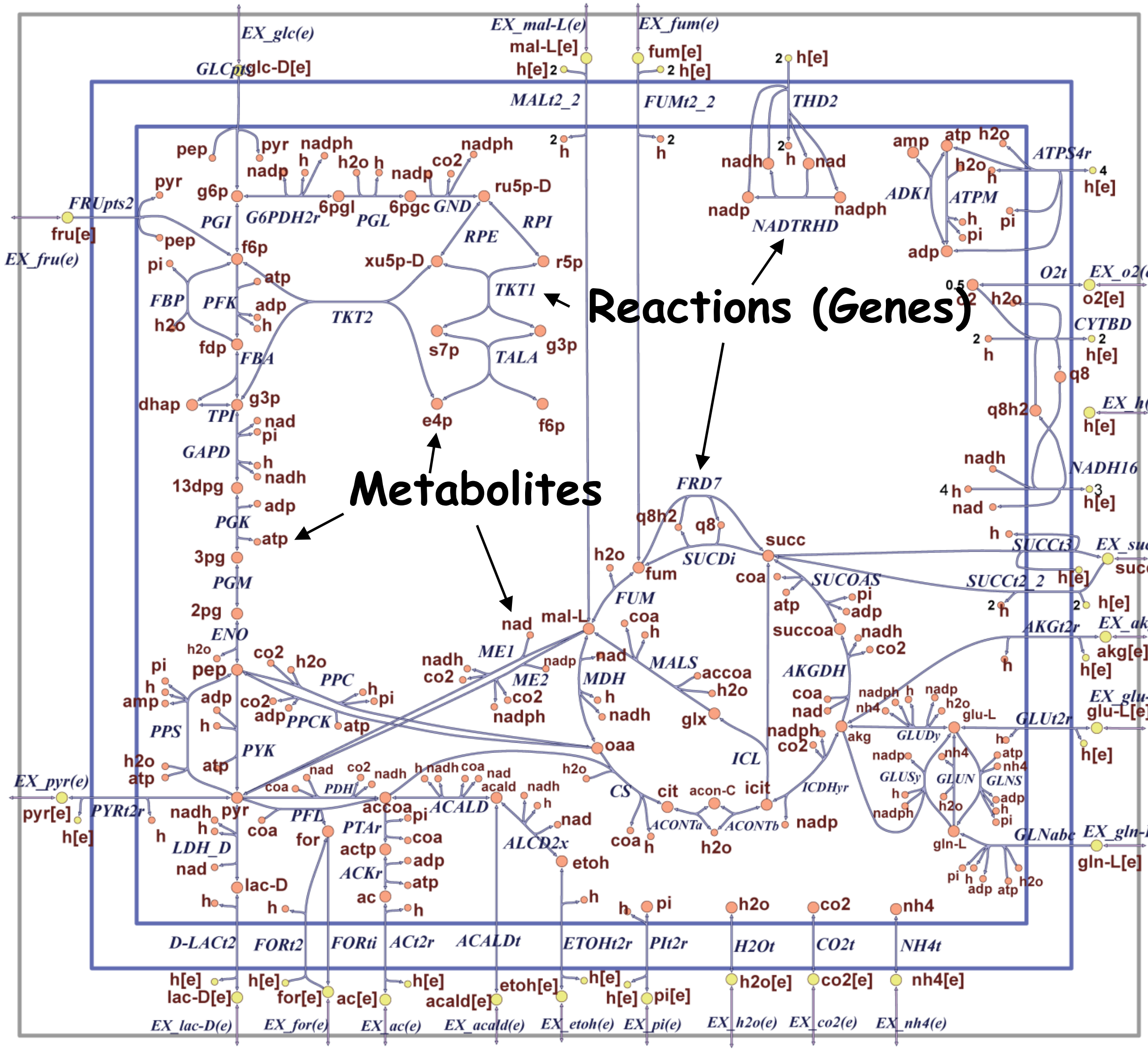
BiGG ID	Organism	Metabolites	Reactions	Genes
e_coli_core	Escherichia coli str. K-12 substr. MG1655	72	95	137
iAB_RBC_283	Homo sapiens	342	469	346
iAF1260	Escherichia coli str. K-12 substr. MG1655	1668	2382	1261
iAF1260b	Escherichia coli str. K-12 substr. MG1655	1668	2388	1261
iAF692	Methanosarcina barkeri str. Fusaro	628	690	692
iAF987	Geobacter metallireducens GS-15	1109	1285	987
iAPECO1_1312	Escherichia coli APEC O1	1942	2736	1313
iAT_PLT_636	Homo sapiens	738	1008	636
iB21_1397	Escherichia coli BL21(DE3)	1943	2742	1337
iBWG_1329	Escherichia coli BW2952	1949	2742	1329
ic_1306	Escherichia coli CFT073	1936	2727	1307
iCHOv1	Cricetulus griseus	4456	6663	1766
iE2348C_1286	Escherichia coli O127:H6 str. E2348/69	1919	2704	1287
iEC042_1314	Escherichia coli 042	1926	2715	1314
iEC55989_1330	Escherichia coli 55989	1953	2757	1330
iECABU_c1320	Escherichia coli ABU 83972	1942	2732	1320
iECB_1328	Escherichia coli B str. REL606	1951	2749	1329
iECBD_1354	Escherichia coli 'BL21-Gold(DE3)pLysS AG'	1952	2749	1354
iECD_1391	Escherichia coli BL21(DE3)	1943	2742	1333

COBRA Models Overview

- COBRApy Toolbox Overview
- ➡ COBRA Models Overview
  - ✓ COBRApy Examples
- Reactions
  - ✓ COBRApy Examples
- Metabolites
  - ✓ COBRApy Examples
- Genes
  - ✓ COBRApy Examples
- Visualization
  - ✓ COBRApy Examples



COBRA Models

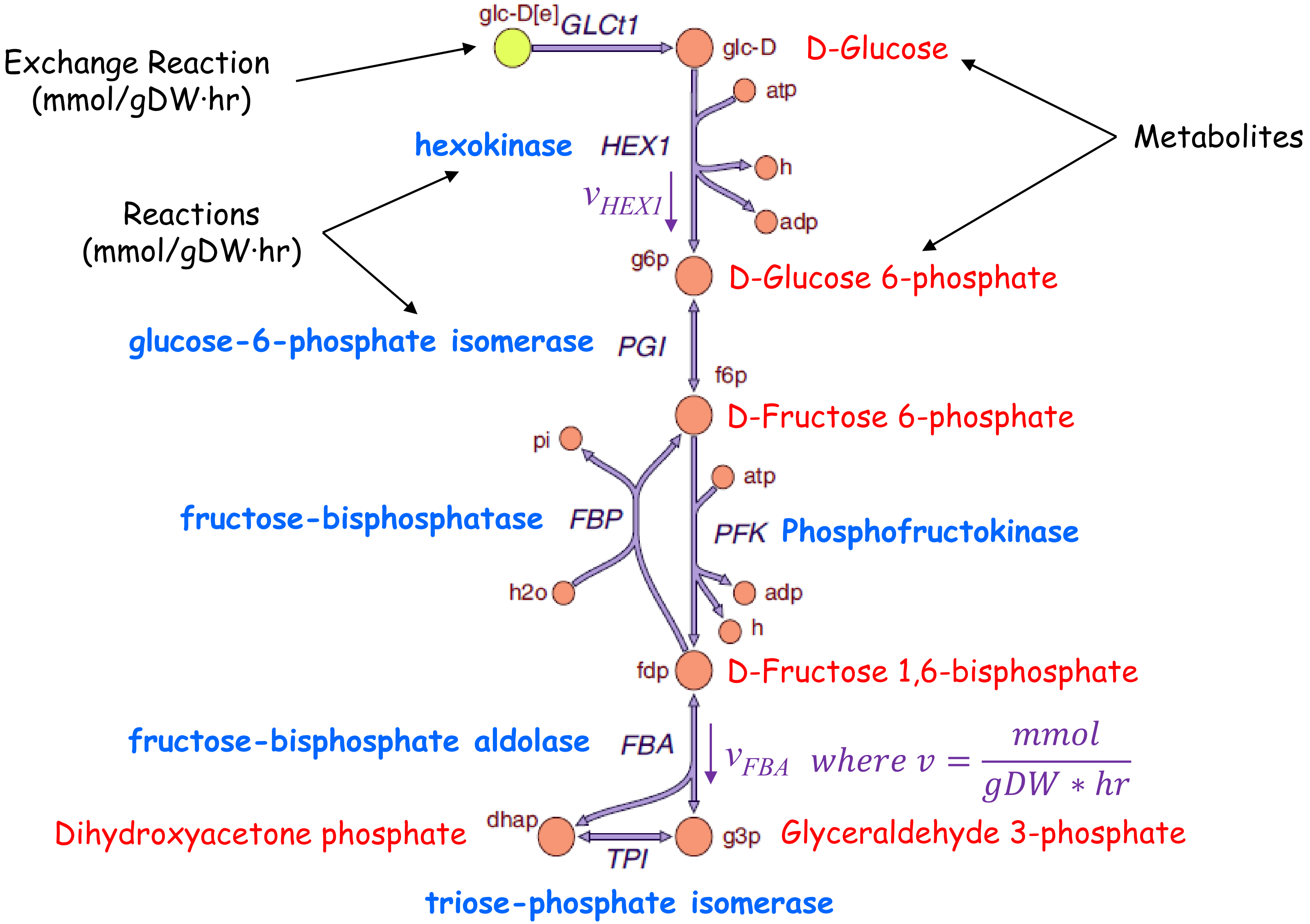


Reactions (Genes)

Metabolites

Orth, J. D., I. Thiele, et al. (2010). "What is flux balance analysis?" Nature biotechnology **28(3): 245-248.**

Metabolic Pathway



Exchange Reaction (mmol/gDW·hr)

Reactions (mmol/gDW·hr)

Metabolites

D-Glucose

hexokinase

v_{HEX1}

D-Glucose 6-phosphate

glucose-6-phosphate isomerase

D-Fructose 6-phosphate

fructose-bisphosphatase

Phosphofructokinase

D-Fructose 1,6-bisphosphate

fructose-bisphosphate aldolase

v_{FBA} $where\ v = \dfrac{mmol}{gDW * hr}$

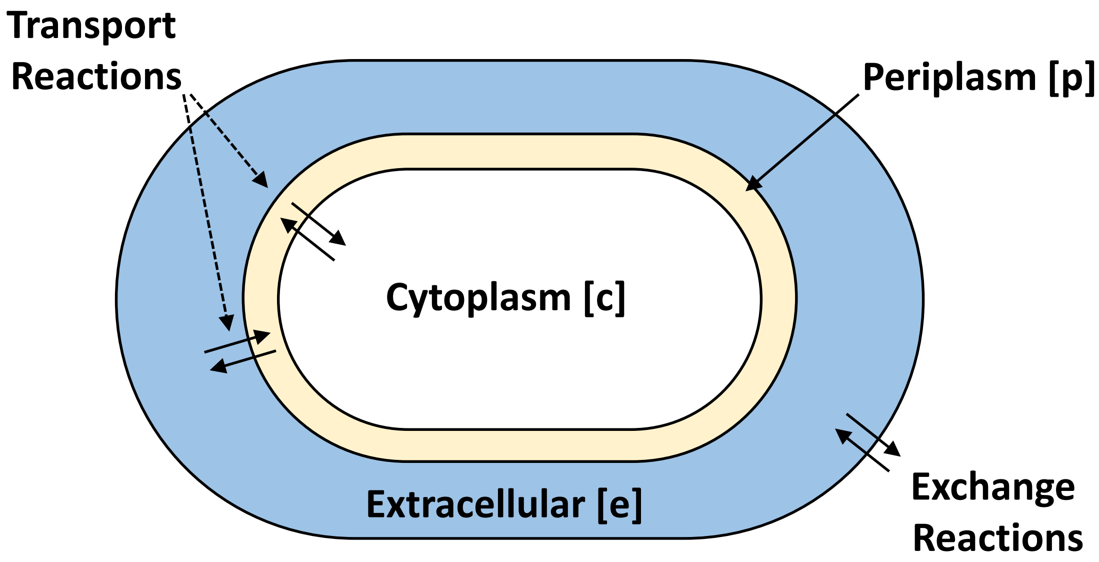
Dihydroxyacetone phosphate

Glyceraldehyde 3-phosphate

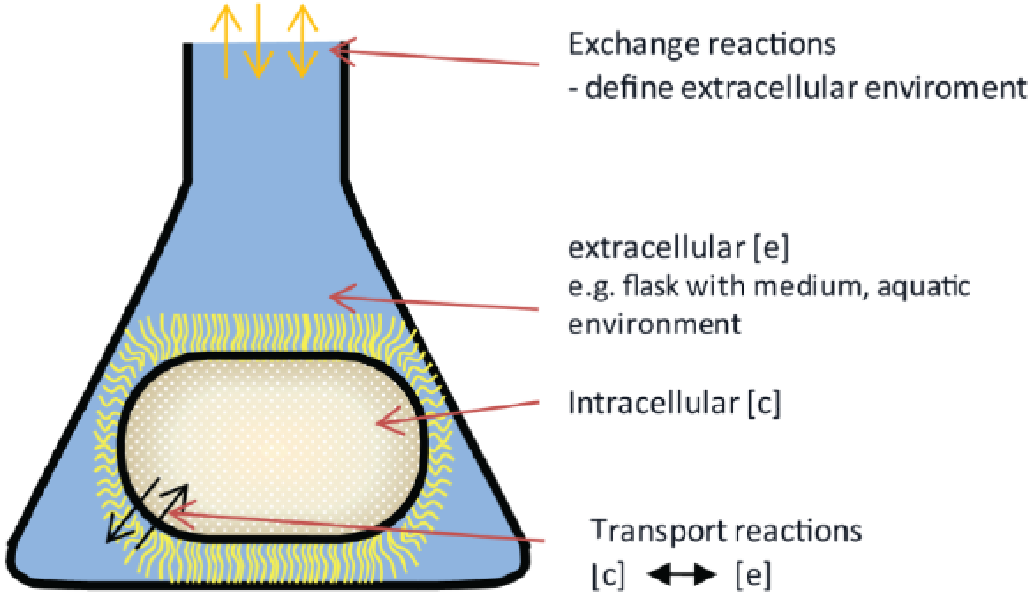
triose-phosphate isomerase

Becker, S. A., et al. (2007). "Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox." Nature protocols **2(3): 727-738.**

System Boundaries:
Exchange & Transport Reactions

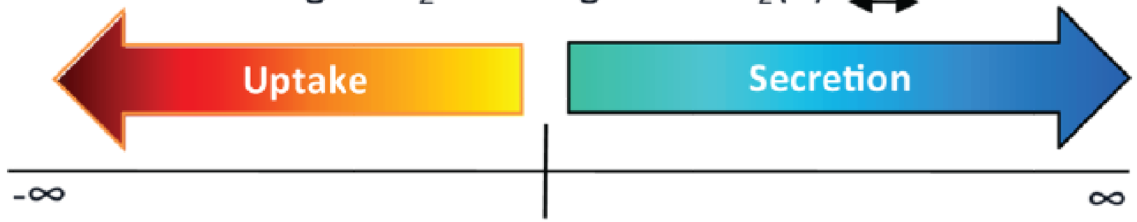


Thiele, I. and B. O. Palsson (2010). "A protocol for generating a high-quality genome-scale metabolic reconstruction." Nature protocols 5(1): 93-121.



Exchange reactions are defined as follows:

e.g.: CO_2 exchange: 1 $CO_2(e)$ ↔

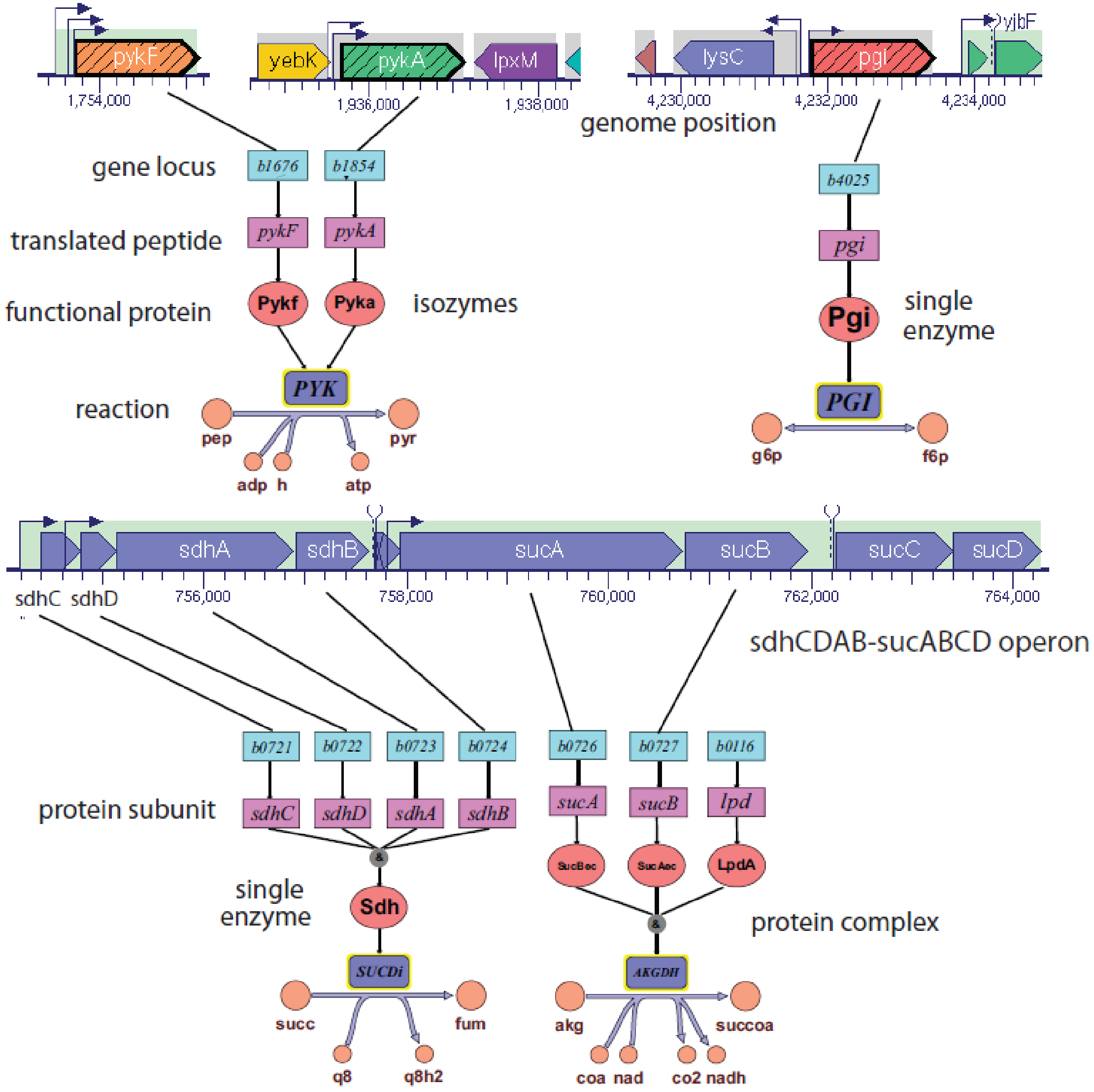


Removal from extracellular environment (hence negative flux value)

Addition to extracellular environment (hence positive flux value)

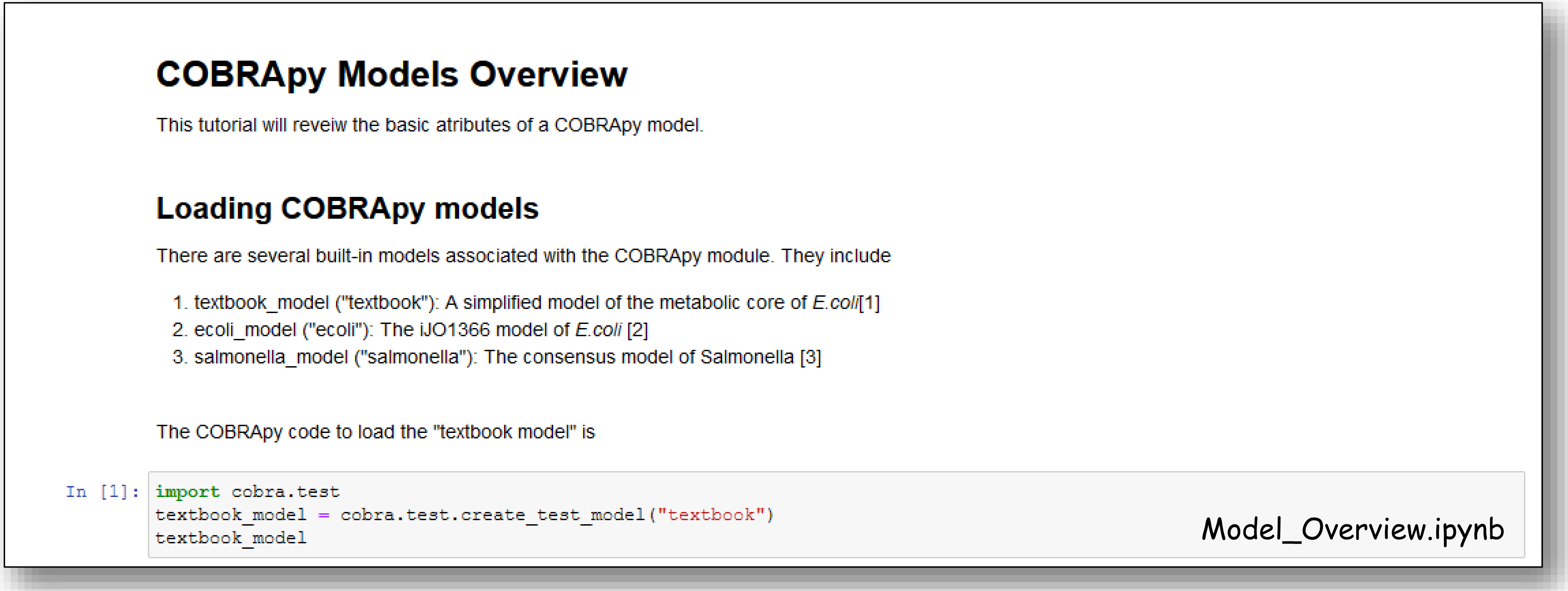
Identifying Metabolic Reactions and Metabolites

(Gene-Protein-Reactions)



Reconstruction and Use of Microbial Metabolic Networks: the Core Escherichia coli Metabolic Model as an Educational Guide by Orth, Fleming, and Palsson (2010)

COBRApy Models Overview



COBRApy Models Overview

This tutorial will reveiw the basic atributes of a COBRApy model.

Loading COBRApy models

There are several built-in models associated with the COBRApy module. They include

1. textbook_model ("textbook"): A simplified model of the metabolic core of *E.coli*[1]
2. ecoli_model ("ecoli"): The iJO1366 model of *E.coli* [2]
3. salmonella_model ("salmonella"): The consensus model of Salmonella [3]

The COBRApy code to load the "textbook model" is

```
In [1]: import cobra.test
        textbook_model = cobra.test.create_test_model("textbook")
        textbook_model
```

Model_Overview.ipynb

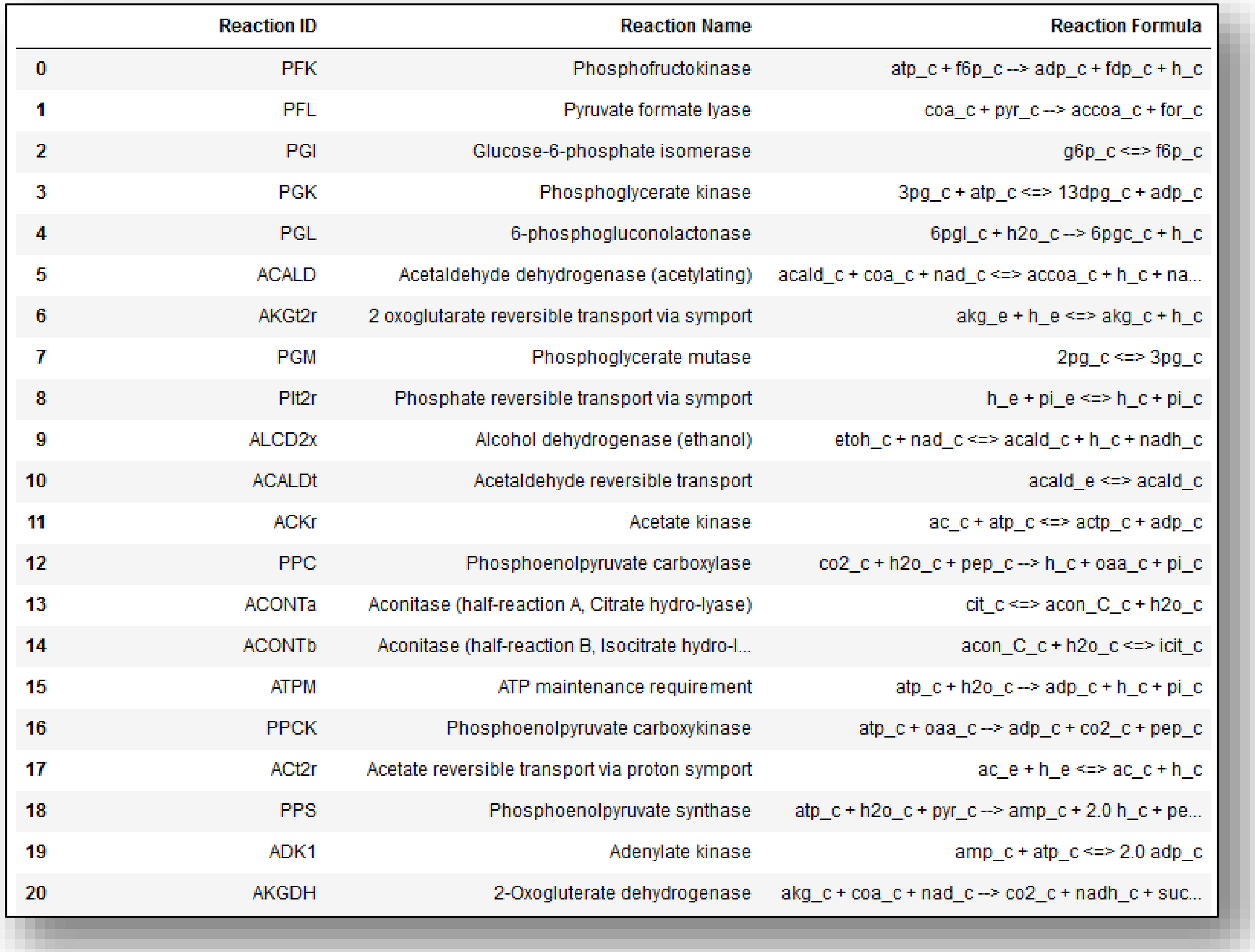
COBRApy Commands for Model Information

(See Model_Overview.ipynb)

- **Loading Models** (import cobra.test)
  - ✓ textbook_model = cobra.test.create_test_model("textbook")
  - ✓ iJO1366_model = cobra.test.create_test_model("ecoli")
  - ✓ salmonella_model = cobra.test.create_test_model("salmonella")
  - ✓ model = cobra.io.load_matlab_model('./e_coli_core.mat') – Loading Matlab model
  - ✓ model = cobra.io.load_json_model('./e_coli_core.json') – Loading JSON model
  - ✓ model = cobra.io.read_sbml_model('./e_coli_core.xml') – Loading model in current directory
- **Model Summary** – “model name” – ex. iJO1366_model
- **Reaction Attributes** – model.reactions?
- **Metabolite Attributes** - model.metabolites?
- **Gene Attributes** - model.genes?
- **Model Compartments** - model.compartments
- **Model Medium** - model.medium
- **Model Sink Reactions, Demand Reactions, Exchange Reactions** – model.sinks, model.demands, model.exchanges
- **Add/Remove Reactions/Metabolites** - model.add_reaction, model.add_metabolite, model.remove_reaction, model.remove_metabolite
- **Objective Function** - print(model.objective)
- **Optimize Flux based on Objective Function** - model.optimize()

COBRA Models Overview

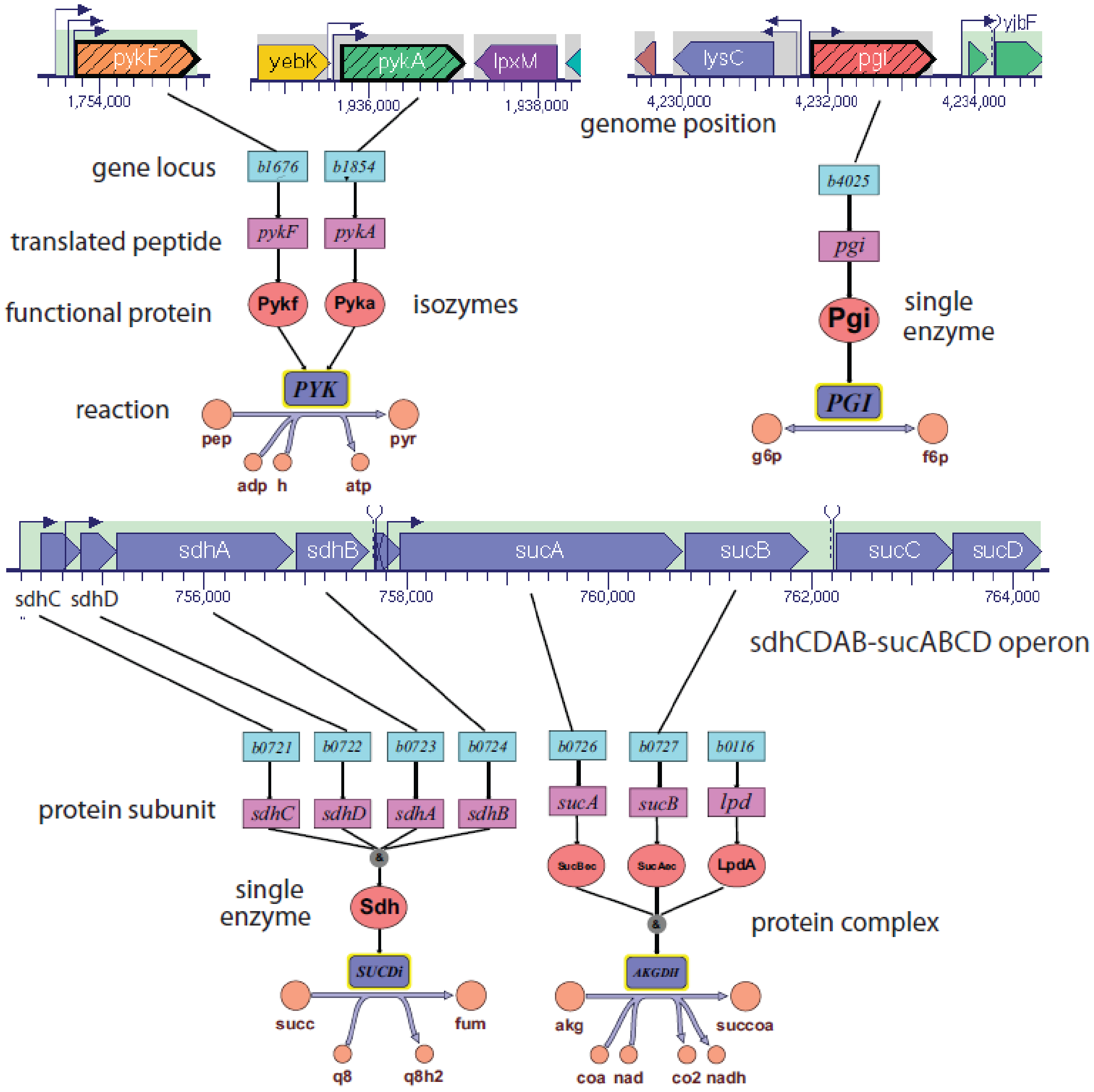
- COBRApy Toolbox Overview
- COBRA Models Overview
  - ✓ COBRApy Examples
- Reactions
  - ✓ COBRApy Examples
- Metabolites
  - ✓ COBRApy Examples
- Genes
  - ✓ COBRApy Examples
- Visualization
  - ✓ COBRApy Examples



	Reaction ID	Reaction Name	Reaction Formula
0	PFK	Phosphofructokinase	atp_c + f6p_c --> adp_c + fdp_c + h_c
1	PFL	Pyruvate formate lyase	coa_c + pyr_c --> accoa_c + for_c
2	PGI	Glucose-6-phosphate isomerase	g6p_c <=> f6p_c
3	PGK	Phosphoglycerate kinase	3pg_c + atp_c <=> 13dpg_c + adp_c
4	PGL	6-phosphogluconolactonase	6pgl_c + h2o_c --> 6pgc_c + h_c
5	ACALD	Acetaldehyde dehydrogenase (acetylating)	acald_c + coa_c + nad_c <=> accoa_c + h_c + na...
6	AKGt2r	2 oxoglutarate reversible transport via symport	akg_e + h_e <=> akg_c + h_c
7	PGM	Phosphoglycerate mutase	2pg_c <=> 3pg_c
8	PIt2r	Phosphate reversible transport via symport	h_e + pi_e <=> h_c + pi_c
9	ALCD2x	Alcohol dehydrogenase (ethanol)	etoh_c + nad_c <=> acald_c + h_c + nadh_c
10	ACALDt	Acetaldehyde reversible transport	acald_e <=> acald_c
11	ACKr	Acetate kinase	ac_c + atp_c <=> actp_c + adp_c
12	PPC	Phosphoenolpyruvate carboxylase	co2_c + h2o_c + pep_c --> h_c + oaa_c + pi_c
13	ACONTa	Aconitase (half-reaction A, Citrate hydro-lyase)	cit_c <=> acon_C_c + h2o_c
14	ACONTb	Aconitase (half-reaction B, Isocitrate hydro-l...	acon_C_c + h2o_c <=> icit_c
15	ATPM	ATP maintenance requirement	atp_c + h2o_c --> adp_c + h_c + pi_c
16	PPCK	Phosphoenolpyruvate carboxykinase	atp_c + oaa_c --> adp_c + co2_c + pep_c
17	ACt2r	Acetate reversible transport via proton symport	ac_e + h_e <=> ac_c + h_c
18	PPS	Phosphoenolpyruvate synthase	atp_c + h2o_c + pyr_c --> amp_c + 2.0 h_c + pe...
19	ADK1	Adenylate kinase	amp_c + atp_c <=> 2.0 adp_c
20	AKGDH	2-Oxogluterate dehydrogenase	akg_c + coa_c + nad_c --> co2_c + nadh_c + suc...

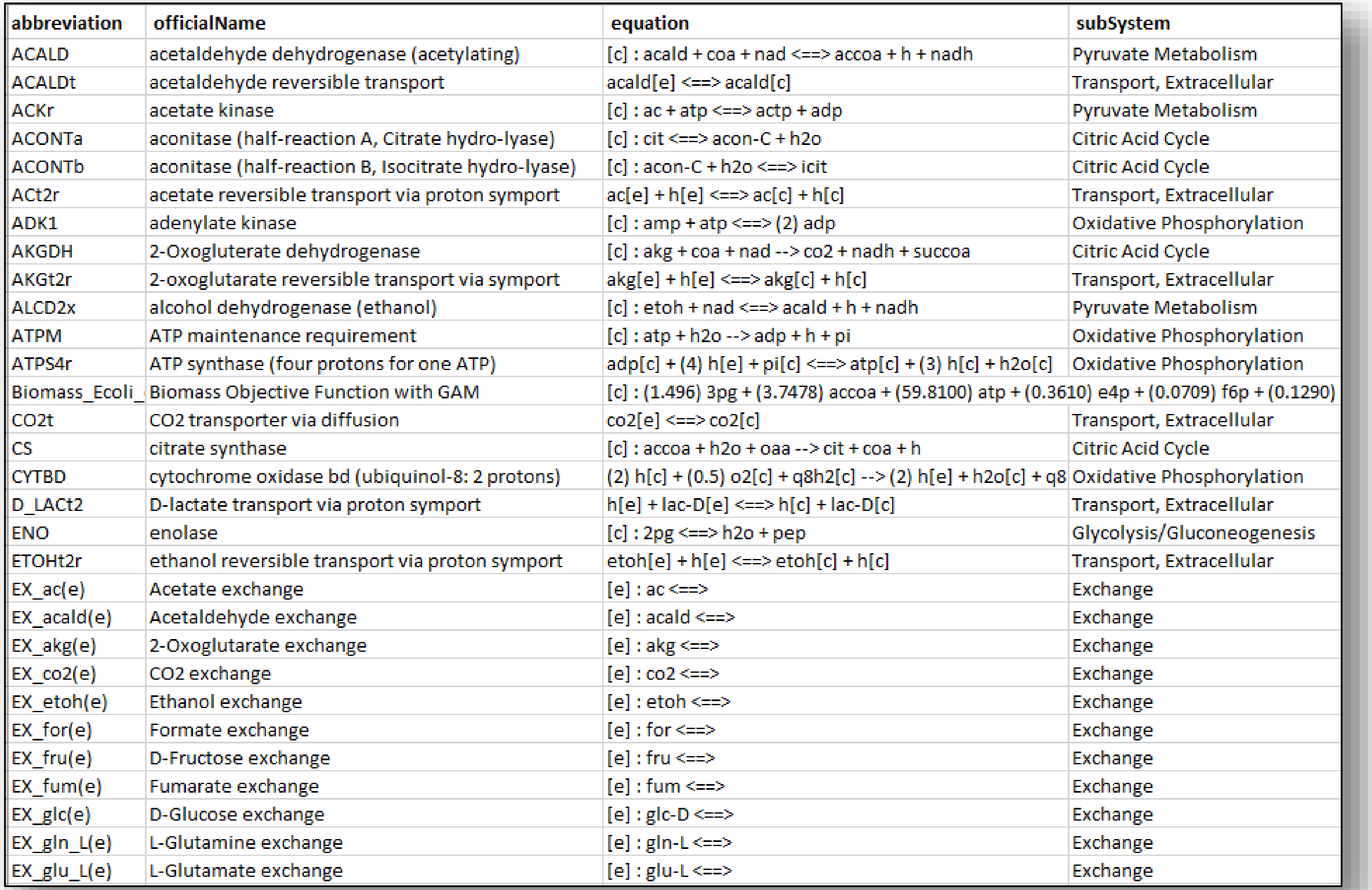
COBRApy Reaction Information

1. Reaction Name
2. Reaction Description
3. Reaction Formula
4. Gene-Reaction Association
5. Genes (Gene Locus)
6. Cellular Subsystem (e.g. Glycolysis)
7. Reaction Direction
8. Flux Lower Bound
9. Flux Upper Bound



Reconstruction and Use of Microbial Metabolic Networks: the Core Escherichia coli Metabolic Model as an Educational Guide by Orth, Fleming, and Palsson (2010)

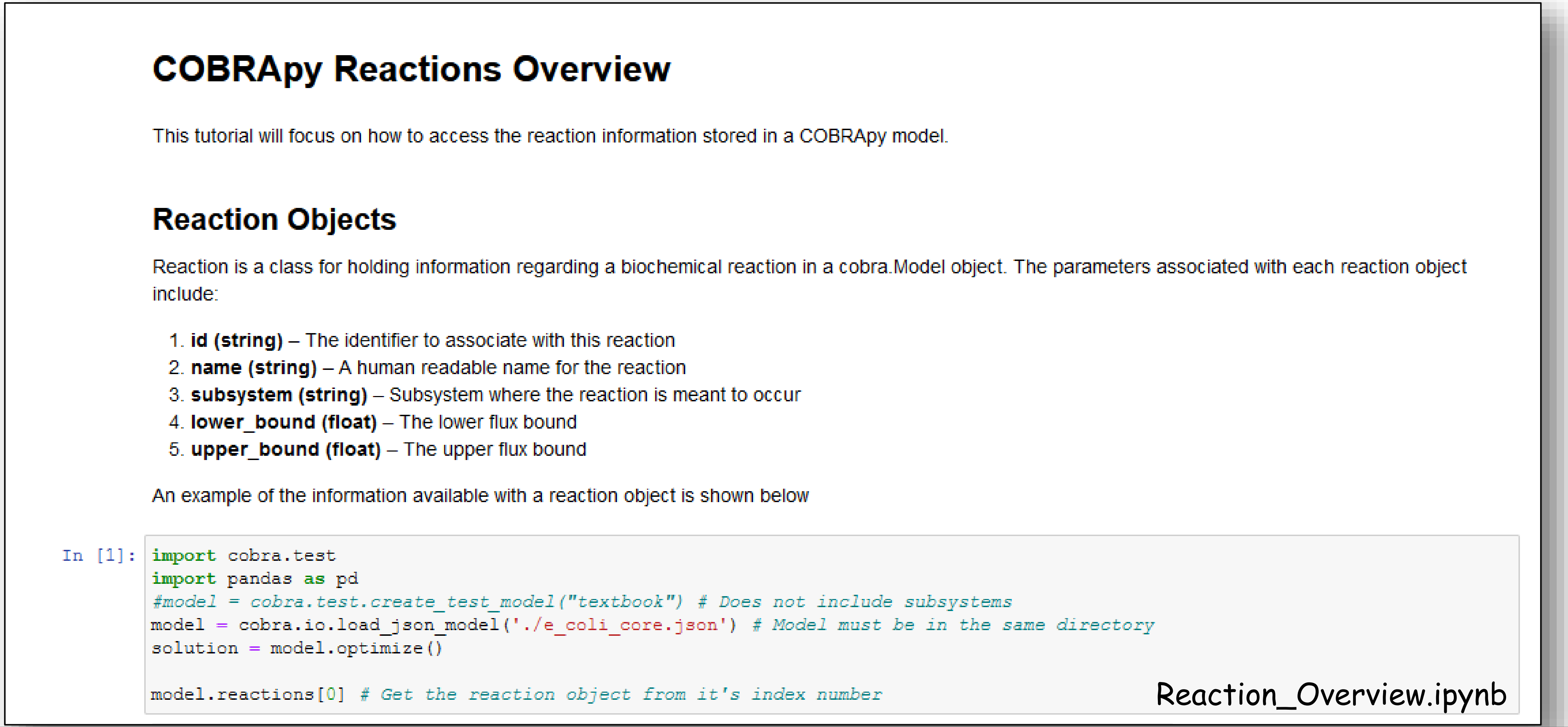
Genome-scale Reconstruction Reactions



abbreviation	officialName	equation	subSystem
ACALD	acetaldehyde dehydrogenase (acetylating)	[c] : acald + coa + nad <==> accoa + h + nadh	Pyruvate Metabolism
ACALDt	acetaldehyde reversible transport	acald[e] <==> acald[c]	Transport, Extracellular
ACKr	acetate kinase	[c] : ac + atp <==> actp + adp	Pyruvate Metabolism
ACONTa	aconitase (half-reaction A, Citrate hydro-lyase)	[c] : cit <==> acon-C + h2o	Citric Acid Cycle
ACONTb	aconitase (half-reaction B, Isocitrate hydro-lyase)	[c] : acon-C + h2o <==> icit	Citric Acid Cycle
ACt2r	acetate reversible transport via proton symport	ac[e] + h[e] <==> ac[c] + h[c]	Transport, Extracellular
ADK1	adenylate kinase	[c] : amp + atp <==> (2) adp	Oxidative Phosphorylation
AKGDH	2-Oxogluterate dehydrogenase	[c] : akg + coa + nad --> co2 + nadh + succoa	Citric Acid Cycle
AKGt2r	2-oxoglutarate reversible transport via symport	akg[e] + h[e] <==> akg[c] + h[c]	Transport, Extracellular
ALCD2x	alcohol dehydrogenase (ethanol)	[c] : etoh + nad <==> acald + h + nadh	Pyruvate Metabolism
ATPM	ATP maintenance requirement	[c] : atp + h2o --> adp + h + pi	Oxidative Phosphorylation
ATPS4r	ATP synthase (four protons for one ATP)	adp[c] + (4) h[e] + pi[c] <==> atp[c] + (3) h[c] + h2o[c]	Oxidative Phosphorylation
Biomass_Ecoli_	Biomass Objective Function with GAM	[c] : (1.496) 3pg + (3.7478) accoa + (59.8100) atp + (0.3610) e4p + (0.0709) f6p + (0.1290)	
CO2t	CO2 transporter via diffusion	co2[e] <==> co2[c]	Transport, Extracellular
CS	citrate synthase	[c] : accoa + h2o + oaa --> cit + coa + h	Citric Acid Cycle
CYTBD	cytochrome oxidase bd (ubiquinol-8: 2 protons)	(2) h[c] + (0.5) o2[c] + q8h2[c] --> (2) h[e] + h2o[c] + q8	Oxidative Phosphorylation
D_LACt2	D-lactate transport via proton symport	h[e] + lac-D[e] <==> h[c] + lac-D[c]	Transport, Extracellular
ENO	enolase	[c] : 2pg <==> h2o + pep	Glycolysis/Gluconeogenesis
ETOHt2r	ethanol reversible transport via proton symport	etoh[e] + h[e] <==> etoh[c] + h[c]	Transport, Extracellular
EX_ac(e)	Acetate exchange	[e] : ac <==>	Exchange
EX_acald(e)	Acetaldehyde exchange	[e] : acald <==>	Exchange
EX_akg(e)	2-Oxoglutarate exchange	[e] : akg <==>	Exchange
EX_co2(e)	CO2 exchange	[e] : co2 <==>	Exchange
EX_etoh(e)	Ethanol exchange	[e] : etoh <==>	Exchange
EX_for(e)	Formate exchange	[e] : for <==>	Exchange
EX_fru(e)	D-Fructose exchange	[e] : fru <==>	Exchange
EX_fum(e)	Fumarate exchange	[e] : fum <==>	Exchange
EX_glc(e)	D-Glucose exchange	[e] : glc-D <==>	Exchange
EX_gln_L(e)	L-Glutamine exchange	[e] : gln-L <==>	Exchange
EX_glu_L(e)	L-Glutamate exchange	[e] : glu-L <==>	Exchange

ecoli_textbook.xls

COBRApy Reactions Overview



COBRApy Reactions Overview

This tutorial will focus on how to access the reaction information stored in a COBRApy model.

Reaction Objects

Reaction is a class for holding information regarding a biochemical reaction in a cobra.Model object. The parameters associated with each reaction object include:

1. **id (string)** – The identifier to associate with this reaction
2. **name (string)** – A human readable name for the reaction
3. **subsystem (string)** – Subsystem where the reaction is meant to occur
4. **lower_bound (float)** – The lower flux bound
5. **upper_bound (float)** – The upper flux bound

An example of the information available with a reaction object is shown below

In [1]:

```
import cobra.test
import pandas as pd
#model = cobra.test.create_test_model("textbook") # Does not include subsystems
model = cobra.io.load_json_model('./e_coli_core.json') # Model must be in the same directory
solution = model.optimize()

model.reactions[0] # Get the reaction object from it's index number
```

Reaction_Overview.ipynb

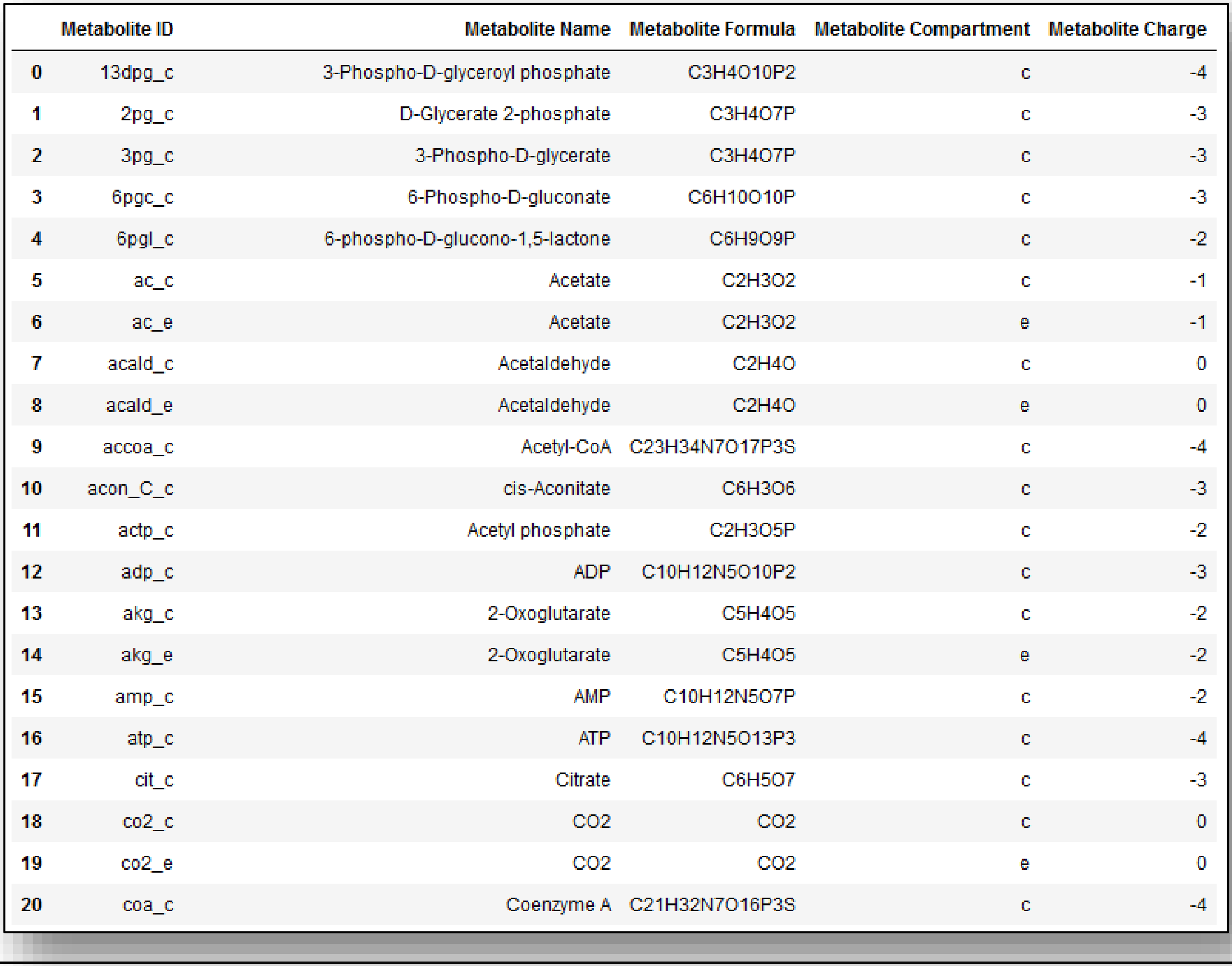
COBRApy Commands for Reaction Information

(See Reaction_Overview.ipynb)

- **Reaction Info** - model.reactions.ID – ex. *model.reactions.PFK*
- **Reaction Summary** - model.reactions.ID.summary() – ex. *model.reactions.PFK.summary()*
- **Reaction Name** - model.reactions.ID.name – ex. *model.reactions.PFK.name*
- **Reaction Formula** - model.reactions.ID.reaction – ex. *model.reactions.PFK.reaction*
- **Reaction Compartments** - model.reactions.ID.compartments – ex. *model.reactions.PFK.compartments*
- **Reaction Subsystem** - model.reactions.ID.subsystem – ex. *model.reactions.PFK.subsystem*
- **Reaction GPR** - model.reactions.ID. *gene_reaction_rule* – ex. *model.reactions.PFK. gene_reaction_rule*
- **Reaction Bounds** - model.reactions.ID.bounds – ex. *model.reactions.PFK.bounds*
- **Reaction Lower Bound** - model.reactions.ID.lower_bound – ex. *model.reactions.PFK.lower_bound*
- **Reaction Upper Bound** - model.reactions.ID.upper_bound – ex. *model.reactions.PFK.upper_bound*
- **Reaction Check Mass Balance** - model.reactions.ID.check_mass_balance() – ex. *model.reactions.PFK. check_mass_balance()*
- **Reaction Flux** - model.reactions.ID.flux – ex. *model.reactions.PFK.flux*
- **Reaction Reduced Cost** - model.reactions.ID.reduced_cost – ex. *model.reactions.PFK.reduced_cost*
- **Reaction Knockout** - model.reactions.ID.knock_out – ex. *model.reactions.PFK.knock-out*

COBRA Models Overview

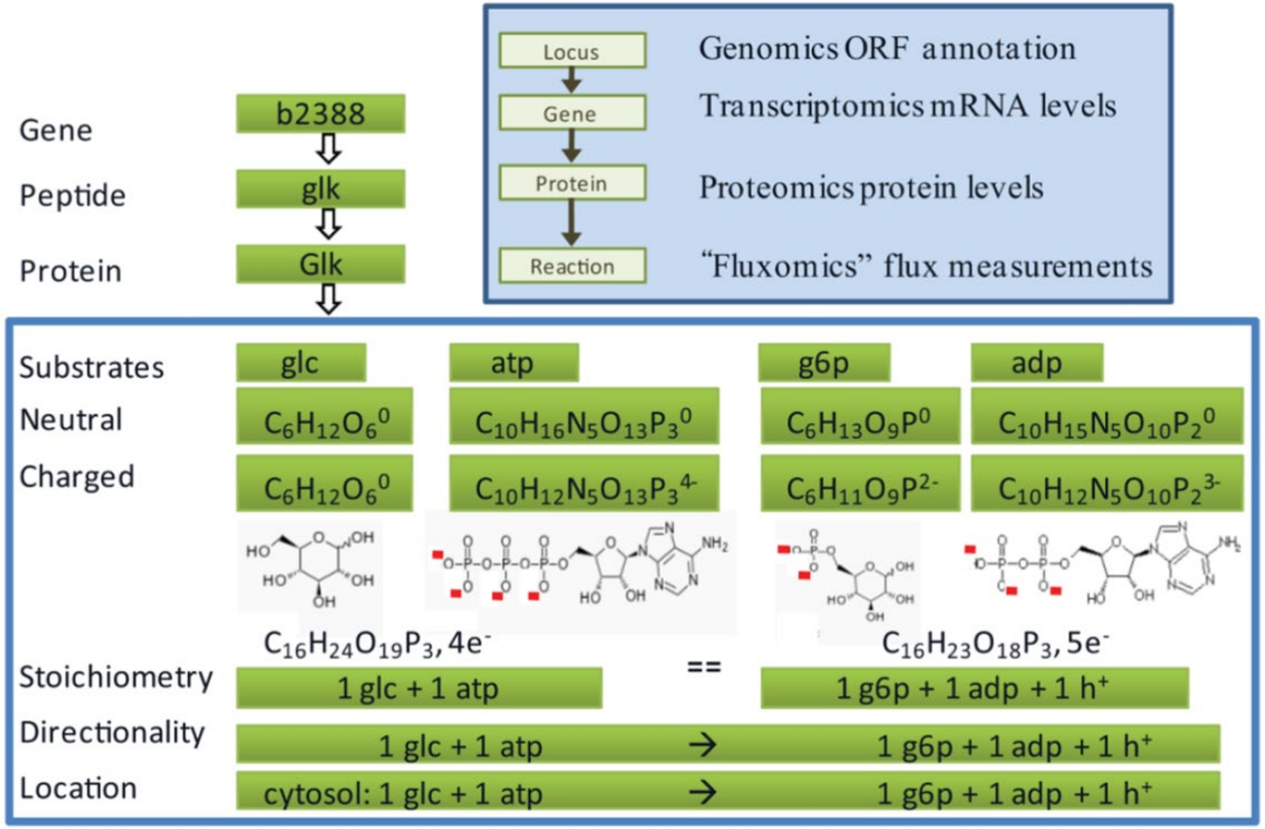
- COBRApy Toolbox Overview
- COBRA Models Overview
  - ✓ COBRApy Examples
- Reactions
  - ✓ COBRApy Examples
- Metabolites
  - ✓ COBRApy Examples
- Genes
  - ✓ COBRApy Examples
- Visualization
  - ✓ COBRApy Examples



	Metabolite ID	Metabolite Name	Metabolite Formula	Metabolite Compartment	Metabolite Charge
0	13dpg_c	3-Phospho-D-glyceroyl phosphate	C3H4O10P2	c	-4
1	2pg_c	D-Glycerate 2-phosphate	C3H4O7P	c	-3
2	3pg_c	3-Phospho-D-glycerate	C3H4O7P	c	-3
3	6pgc_c	6-Phospho-D-gluconate	C6H10O10P	c	-3
4	6pgl_c	6-phospho-D-glucono-1,5-lactone	C6H9O9P	c	-2
5	ac_c	Acetate	C2H3O2	c	-1
6	ac_e	Acetate	C2H3O2	e	-1
7	acald_c	Acetaldehyde	C2H4O	c	0
8	acald_e	Acetaldehyde	C2H4O	e	0
9	accoa_c	Acetyl-CoA	C23H34N7O17P3S	c	-4
10	acon_C_c	cis-Aconitate	C6H3O6	c	-3
11	actp_c	Acetyl phosphate	C2H3O5P	c	-2
12	adp_c	ADP	C10H12N5O10P2	c	-3
13	akg_c	2-Oxoglutarate	C5H4O5	c	-2
14	akg_e	2-Oxoglutarate	C5H4O5	e	-2
15	amp_c	AMP	C10H12N5O7P	c	-2
16	atp_c	ATP	C10H12N5O13P3	c	-4
17	cit_c	Citrate	C6H5O7	c	-3
18	co2_c	CO2	CO2	c	0
19	co2_e	CO2	CO2	e	0
20	coa_c	Coenzyme A	C21H32N7O16P3S	c	-4

COBRApy Metabolite Information

1. Metabolite Name
2. Metabolite Description
3. Metabolite Formula
4. Metabolite Charge
5. Metabolite Compartment



Locus → Gene → Protein → Reaction

Genomics ORF annotation

Transcriptomics mRNA levels

Proteomics protein levels

“Fluxomics” flux measurements

Gene: b2388 ⇩

Peptide: glk ⇩

Protein: Glk ⇩

Substrates	glc	atp	g6p	adp
Neutral	$C_6H_{12}O_6{}^0$	$C_{10}H_{16}N_5O_{13}P_3{}^0$	$C_6H_{13}O_9P^0$	$C_{10}H_{15}N_5O_{10}P_2{}^0$
Charged	$C_6H_{12}O_6{}^0$	$C_{10}H_{12}N_5O_{13}P_3{}^{4-}$	$C_6H_{11}O_9P^{2-}$	$C_{10}H_{12}N_5O_{10}P_2{}^{3-}$

$C_{16}H_{24}O_{19}P_3, 4e^-$ == $C_{16}H_{23}O_{18}P_3, 5e^-$

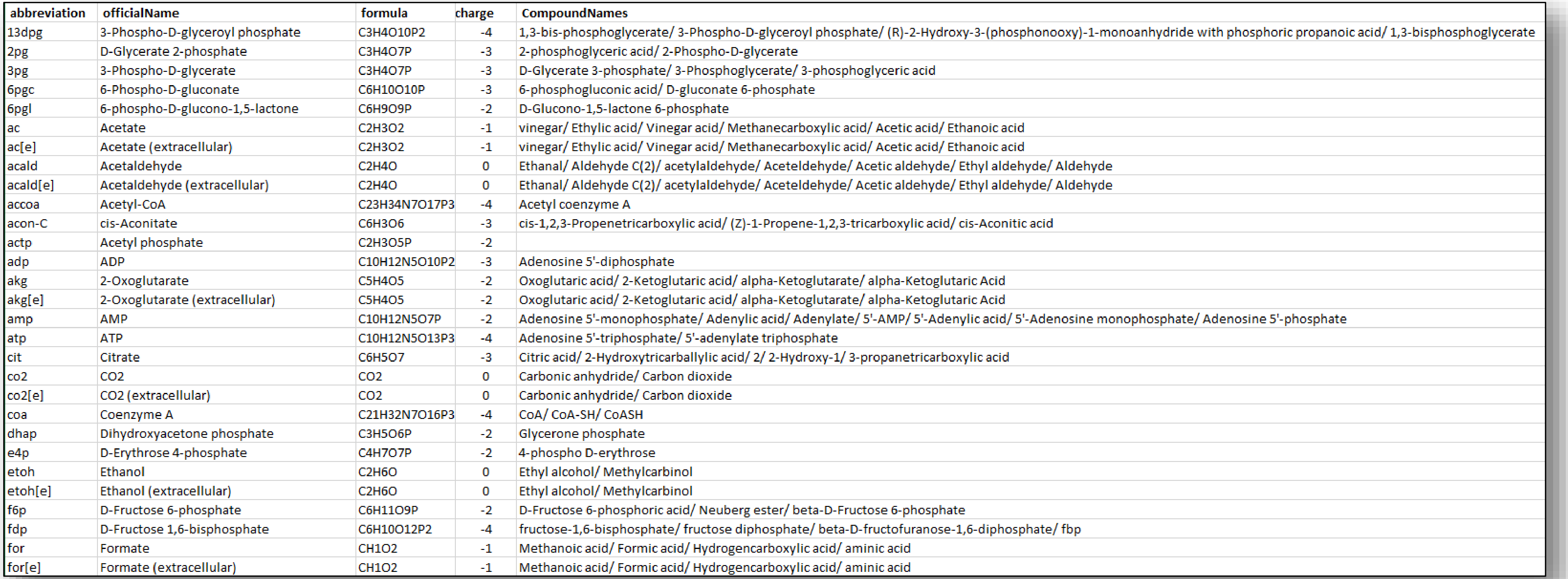
Stoichiometry: 1 glc + 1 atp == 1 g6p + 1 adp + 1 h^+

Directionality: 1 glc + 1 atp → 1 g6p + 1 adp + 1 h^+

Location: cytosol: 1 glc + 1 atp → 1 g6p + 1 adp + 1 h^+

Thiele, I. and B. O. Palsson (2010). "A protocol for generating a high-quality genome-scale metabolic reconstruction." Nature protocols 5(1): 93-121.

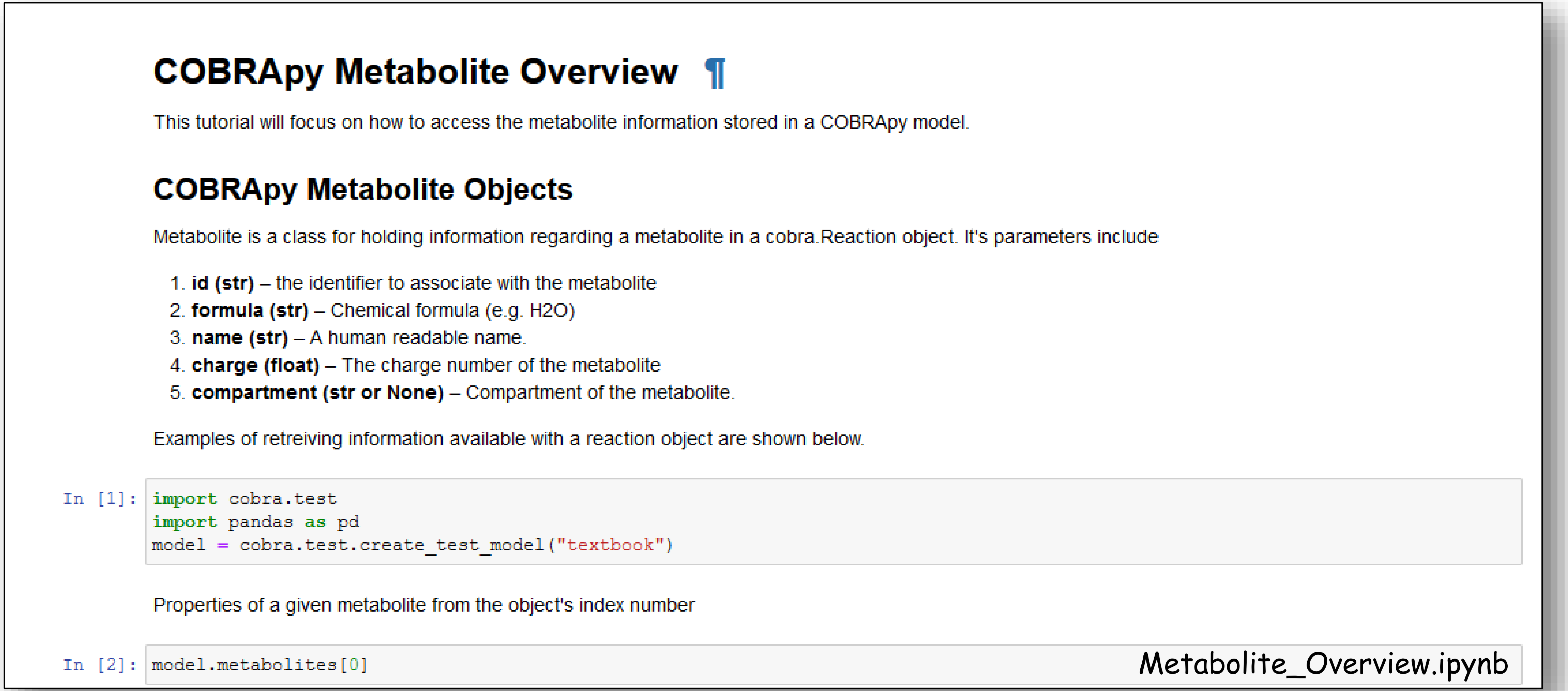
Genome-scale Reconstruction Metabolites



abbreviation	officialName	formula	charge	CompoundNames
13dpg	3-Phospho-D-glyceroyl phosphate	C3H4O10P2	-4	1,3-bis-phosphoglycerate/ 3-Phospho-D-glyceroyl phosphate/ (R)-2-Hydroxy-3-(phosphonooxy)-1-monoanhydride with phosphoric propanoic acid/ 1,3-bisphosphoglycerate
2pg	D-Glycerate 2-phosphate	C3H4O7P	-3	2-phosphoglyceric acid/ 2-Phospho-D-glycerate
3pg	3-Phospho-D-glycerate	C3H4O7P	-3	D-Glycerate 3-phosphate/ 3-Phosphoglycerate/ 3-phosphoglyceric acid
6pgc	6-Phospho-D-gluconate	C6H10O10P	-3	6-phosphogluconic acid/ D-gluconate 6-phosphate
6pgl	6-phospho-D-glucono-1,5-lactone	C6H9O9P	-2	D-Glucono-1,5-lactone 6-phosphate
ac	Acetate	C2H3O2	-1	vinegar/ Ethylic acid/ Vinegar acid/ Methanecarboxylic acid/ Acetic acid/ Ethanoic acid
ac[e]	Acetate (extracellular)	C2H3O2	-1	vinegar/ Ethylic acid/ Vinegar acid/ Methanecarboxylic acid/ Acetic acid/ Ethanoic acid
acald	Acetaldehyde	C2H4O	0	Ethanal/ Aldehyde C(2)/ acetylaldehyde/ Aceteldehyde/ Acetic aldehyde/ Ethyl aldehyde/ Aldehyde
acald[e]	Acetaldehyde (extracellular)	C2H4O	0	Ethanal/ Aldehyde C(2)/ acetylaldehyde/ Aceteldehyde/ Acetic aldehyde/ Ethyl aldehyde/ Aldehyde
accoa	Acetyl-CoA	C23H34N7O17P3	-4	Acetyl coenzyme A
acon-C	cis-Aconitate	C6H3O6	-3	cis-1,2,3-Propenetricarboxylic acid/ (Z)-1-Propene-1,2,3-tricarboxylic acid/ cis-Aconitic acid
actp	Acetyl phosphate	C2H3O5P	-2	
adp	ADP	C10H12N5O10P2	-3	Adenosine 5'-diphosphate
akg	2-Oxoglutarate	C5H4O5	-2	Oxoglutaric acid/ 2-Ketoglutaric acid/ alpha-Ketoglutarate/ alpha-Ketoglutaric Acid
akg[e]	2-Oxoglutarate (extracellular)	C5H4O5	-2	Oxoglutaric acid/ 2-Ketoglutaric acid/ alpha-Ketoglutarate/ alpha-Ketoglutaric Acid
amp	AMP	C10H12N5O7P	-2	Adenosine 5'-monophosphate/ Adenylic acid/ Adenylate/ 5'-AMP/ 5'-Adenylic acid/ 5'-Adenosine monophosphate/ Adenosine 5'-phosphate
atp	ATP	C10H12N5O13P3	-4	Adenosine 5'-triphosphate/ 5'-adenylate triphosphate
cit	Citrate	C6H5O7	-3	Citric acid/ 2-Hydroxytricarballylic acid/ 2/ 2-Hydroxy-1/ 3-propanetricarboxylic acid
co2	CO2	CO2	0	Carbonic anhydride/ Carbon dioxide
co2[e]	CO2 (extracellular)	CO2	0	Carbonic anhydride/ Carbon dioxide
coa	Coenzyme A	C21H32N7O16P3	-4	CoA/ CoA-SH/ CoASH
dhap	Dihydroxyacetone phosphate	C3H5O6P	-2	Glycerone phosphate
e4p	D-Erythrose 4-phosphate	C4H7O7P	-2	4-phospho D-erythrose
etoh	Ethanol	C2H6O	0	Ethyl alcohol/ Methylcarbinol
etoh[e]	Ethanol (extracellular)	C2H6O	0	Ethyl alcohol/ Methylcarbinol
f6p	D-Fructose 6-phosphate	C6H11O9P	-2	D-Fructose 6-phosphoric acid/ Neuberg ester/ beta-D-Fructose 6-phosphate
fdp	D-Fructose 1,6-bisphosphate	C6H10O12P2	-4	fructose-1,6-bisphosphate/ fructose diphosphate/ beta-D-fructofuranose-1,6-diphosphate/ fbp
for	Formate	CH1O2	-1	Methanoic acid/ Formic acid/ Hydrogencarboxylic acid/ aminic acid
for[e]	Formate (extracellular)	CH1O2	-1	Methanoic acid/ Formic acid/ Hydrogencarboxylic acid/ aminic acid

ecoli_textbook.xls

COBRApy Metabolite Overview



COBRApy Metabolite Overview ¶

This tutorial will focus on how to access the metabolite information stored in a COBRApy model.

COBRApy Metabolite Objects

Metabolite is a class for holding information regarding a metabolite in a cobra.Reaction object. It's parameters include

1. **id (str)** – the identifier to associate with the metabolite
2. **formula (str)** – Chemical formula (e.g. H2O)
3. **name (str)** – A human readable name.
4. **charge (float)** – The charge number of the metabolite
5. **compartment (str or None)** – Compartment of the metabolite.

Examples of retreiving information available with a reaction object are shown below.

```
In [1]: import cobra.test
        import pandas as pd
        model = cobra.test.create_test_model("textbook")
```

Properties of a given metabolite from the object's index number

```
In [2]: model.metabolites[0]
```

Metabolite_Overview.ipynb

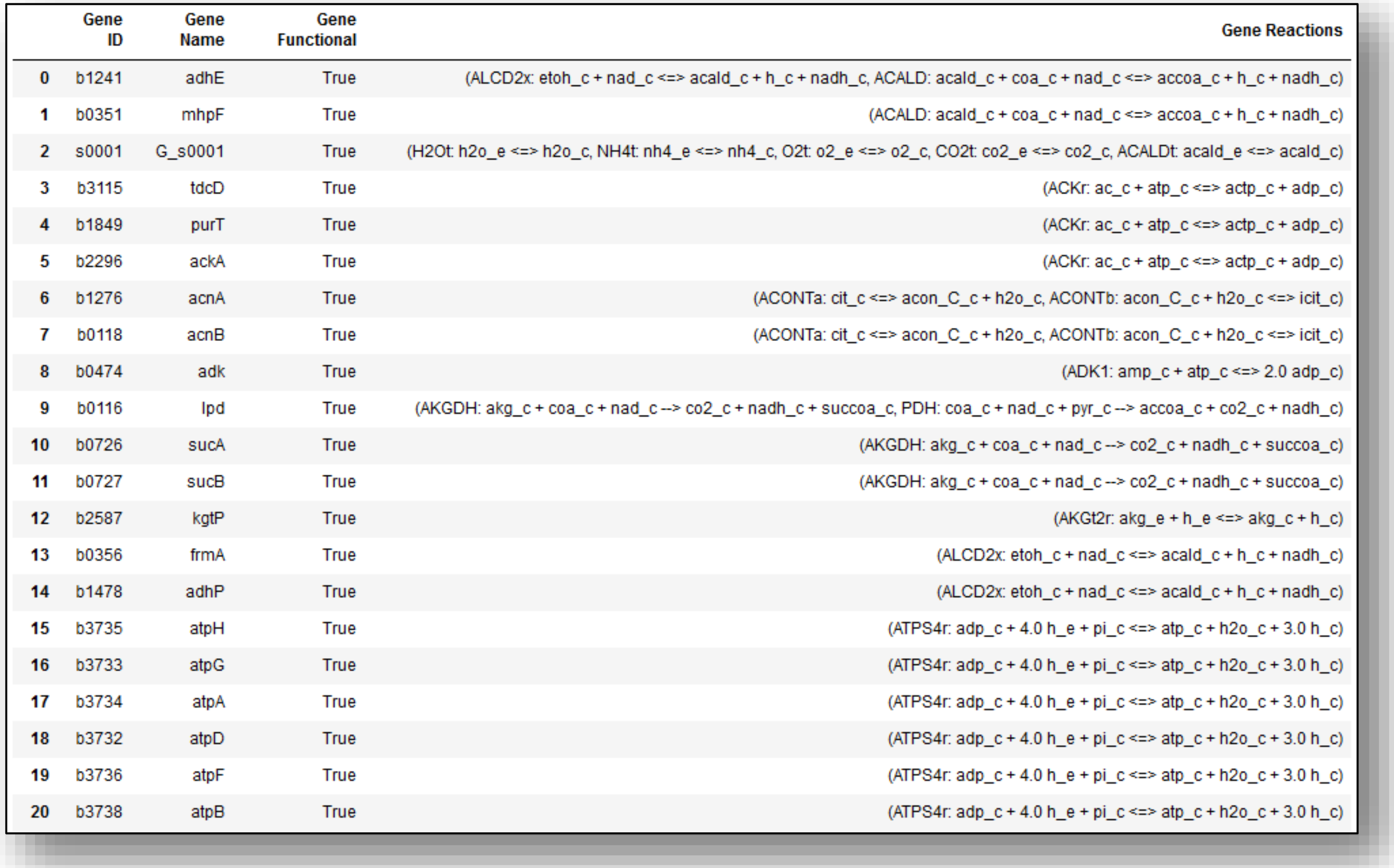
COBRApy Commands for Metabolite Information

(See Metabolite_Overview.ipynb)

- **Metabolite Info by Index Number** - model.metabolites[index]– ex. *model.metabolites[1]*
- **Metabolite Info by ID** - model.metabolites.get_by_ID(‘metabolite ID’) – ex. *model.metabolites.get_by_ID(’13dpg_c’)*
- **Metabolite Name** - model.metabolites.get_by_ID(‘metabolite ID’).name – ex. *model.metabolites.get_by_ID(’13dpg_c’).name*
- **Metabolite Formula** - model.metabolites.get_by_ID(‘metabolite ID’).formula – ex. *model.metabolites.get_by_ID(’13dpg_c’).formula*
- **Metabolite Elements** - model.metabolites.get_by_ID(‘metabolite ID’).elements – ex. *model.metabolites.get_by_ID(’13dpg_c’).elements*
- **Metabolite Formula Weight** - model.metabolites.get_by_ID(‘metabolite ID’).formula_weight – ex. *model.metabolites.get_by_ID(’13dpg_c’).formula_weight*
- **Metabolite Compartments** - model.metabolites.get_by_ID(‘metabolite ID’).compartment– ex. *model.metabolites.get_by_ID(’13dpg_c’).compartment*
- **Metabolite Charge** - model.metabolites.get_by_ID(‘metabolite ID’).charge – ex. *model.metabolites.get_by_ID(’13dpg_c’).charge*
- **Metabolite Summary** - model.metabolites.get_by_ID(‘metabolite ID’).summary() – ex. *model.metabolites.get_by_ID(’13dpg_c’).summary()*
- **Metabolite Reactions** - model.metabolites.get_by_ID(‘metabolite ID’).reactions – ex. *model.metabolites.get_by_ID(’13dpg_c’).reactions*
- **Metabolite Shadow Prices** - model.metabolites.get_by_ID(‘metabolite ID’).shadow_price – ex. *model.metabolites.get_by_ID(’13dpg_c’).shadow_price*

COBRA Models Overview

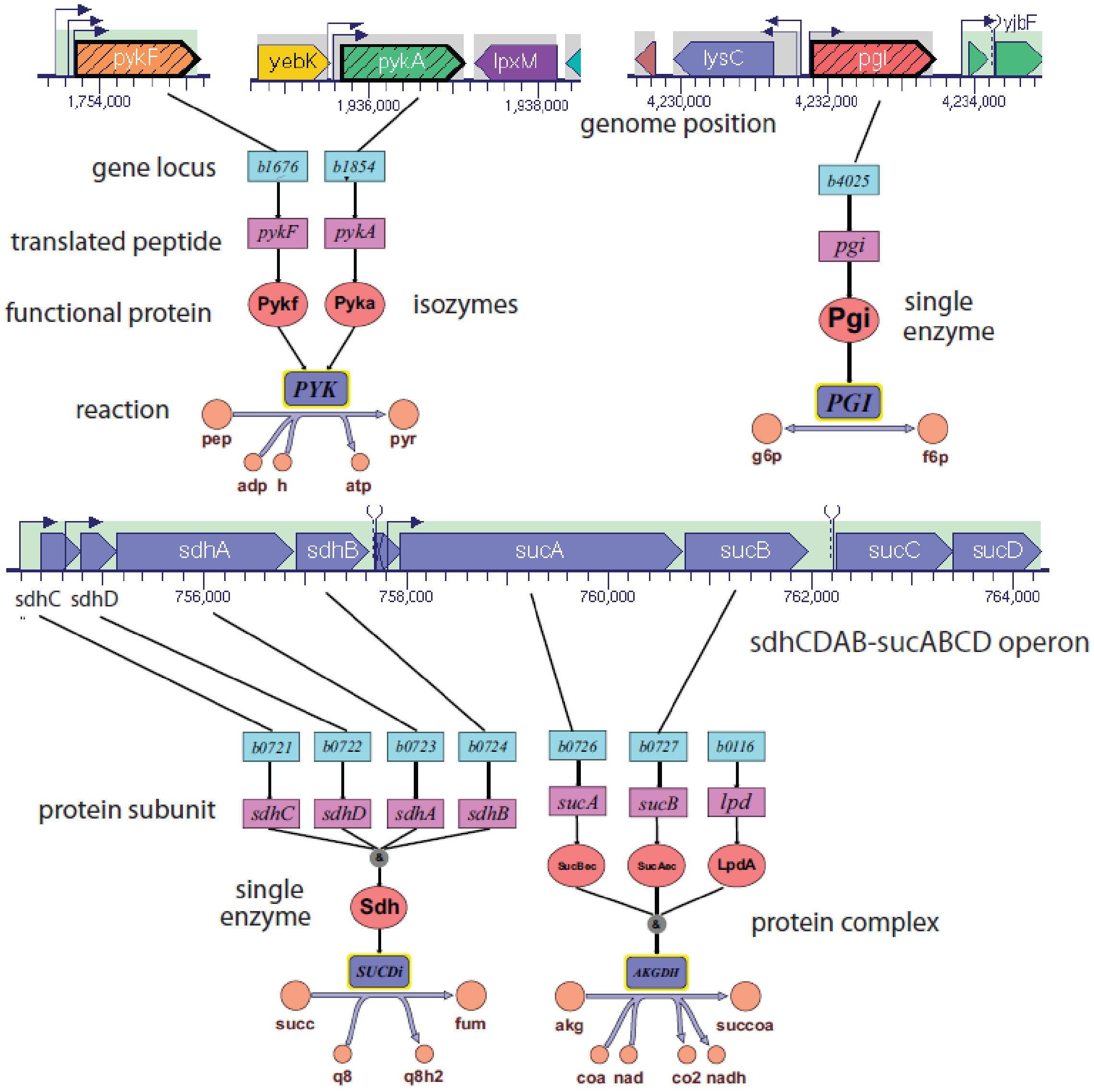
- COBRApy Toolbox Overview
- COBRA Models Overview
  - ✓ COBRApy Examples
- Reactions
  - ✓ COBRApy Examples
- Metabolites
  - ✓ COBRApy Examples
- ➡ Genes
  - ✓ COBRApy Examples
- Visualization
  - ✓ COBRApy Examples



	Gene ID	Gene Name	Gene Functional	Gene Reactions
0	b1241	adhE	True	(ALCD2x: etoh_c + nad_c <=> acald_c + h_c + nadh_c, ACALD: acald_c + coa_c + nad_c <=> accoa_c + h_c + nadh_c)
1	b0351	mhpF	True	(ACALD: acald_c + coa_c + nad_c <=> accoa_c + h_c + nadh_c)
2	s0001	G_s0001	True	(H2Ot: h2o_e <=> h2o_c, NH4t: nh4_e <=> nh4_c, O2t: o2_e <=> o2_c, CO2t: co2_e <=> co2_c, ACALDt: acald_e <=> acald_c)
3	b3115	tdcD	True	(ACKr: ac_c + atp_c <=> actp_c + adp_c)
4	b1849	purT	True	(ACKr: ac_c + atp_c <=> actp_c + adp_c)
5	b2296	ackA	True	(ACKr: ac_c + atp_c <=> actp_c + adp_c)
6	b1276	acnA	True	(ACONTa: cit_c <=> acon_C_c + h2o_c, ACONTb: acon_C_c + h2o_c <=> icit_c)
7	b0118	acnB	True	(ACONTa: cit_c <=> acon_C_c + h2o_c, ACONTb: acon_C_c + h2o_c <=> icit_c)
8	b0474	adk	True	(ADK1: amp_c + atp_c <=> 2.0 adp_c)
9	b0116	lpd	True	(AKGDH: akg_c + coa_c + nad_c --> co2_c + nadh_c + succoa_c, PDH: coa_c + nad_c + pyr_c --> accoa_c + co2_c + nadh_c)
10	b0726	sucA	True	(AKGDH: akg_c + coa_c + nad_c --> co2_c + nadh_c + succoa_c)
11	b0727	sucB	True	(AKGDH: akg_c + coa_c + nad_c --> co2_c + nadh_c + succoa_c)
12	b2587	kgtP	True	(AKGt2r: akg_e + h_e <=> akg_c + h_c)
13	b0356	frmA	True	(ALCD2x: etoh_c + nad_c <=> acald_c + h_c + nadh_c)
14	b1478	adhP	True	(ALCD2x: etoh_c + nad_c <=> acald_c + h_c + nadh_c)
15	b3735	atpH	True	(ATPS4r: adp_c + 4.0 h_e + pi_c <=> atp_c + h2o_c + 3.0 h_c)
16	b3733	atpG	True	(ATPS4r: adp_c + 4.0 h_e + pi_c <=> atp_c + h2o_c + 3.0 h_c)
17	b3734	atpA	True	(ATPS4r: adp_c + 4.0 h_e + pi_c <=> atp_c + h2o_c + 3.0 h_c)
18	b3732	atpD	True	(ATPS4r: adp_c + 4.0 h_e + pi_c <=> atp_c + h2o_c + 3.0 h_c)
19	b3736	atpF	True	(ATPS4r: adp_c + 4.0 h_e + pi_c <=> atp_c + h2o_c + 3.0 h_c)
20	b3738	atpB	True	(ATPS4r: adp_c + 4.0 h_e + pi_c <=> atp_c + h2o_c + 3.0 h_c)

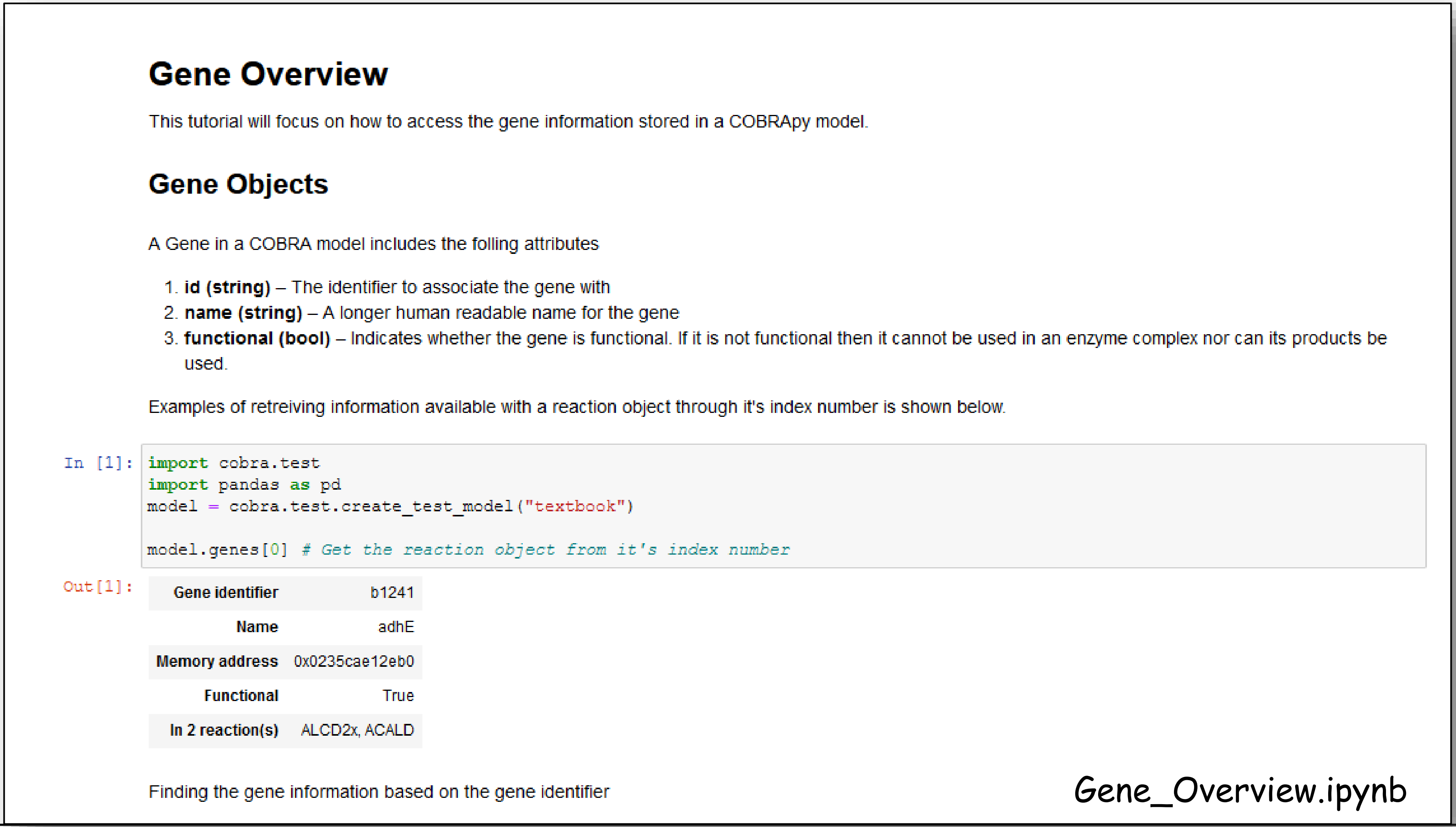
COBRApy Gene Information

1. Gene ID (Gene Locus)
2. Gene Name
3. Gene Description
4. Gene Functional
5. Associated Reactions



Reconstruction and Use of Microbial Metabolic Networks: the Core Escherichia coli Metabolic Model as an Educational Guide by Orth, Fleming, and Palsson (2010)

COBRApy Gene Overview



Gene Overview

This tutorial will focus on how to access the gene information stored in a COBRApy model.

Gene Objects

A Gene in a COBRA model includes the folling attributes

1. **id (string)** – The identifier to associate the gene with
2. **name (string)** – A longer human readable name for the gene
3. **functional (bool)** – Indicates whether the gene is functional. If it is not functional then it cannot be used in an enzyme complex nor can its products be used.

Examples of retreiving information available with a reaction object through it's index number is shown below.

```
In [1]: import cobra.test
        import pandas as pd
        model = cobra.test.create_test_model("textbook")

        model.genes[0] # Get the reaction object from it's index number
```

```
Out[1]:    Gene identifier            b1241
                      Name             adhE
            Memory address   0x0235cae12eb0
                Functional             True
          In 2 reaction(s)      ALCD2x, ACALD
```

Finding the gene information based on the gene identifier

Gene_Overview.ipynb

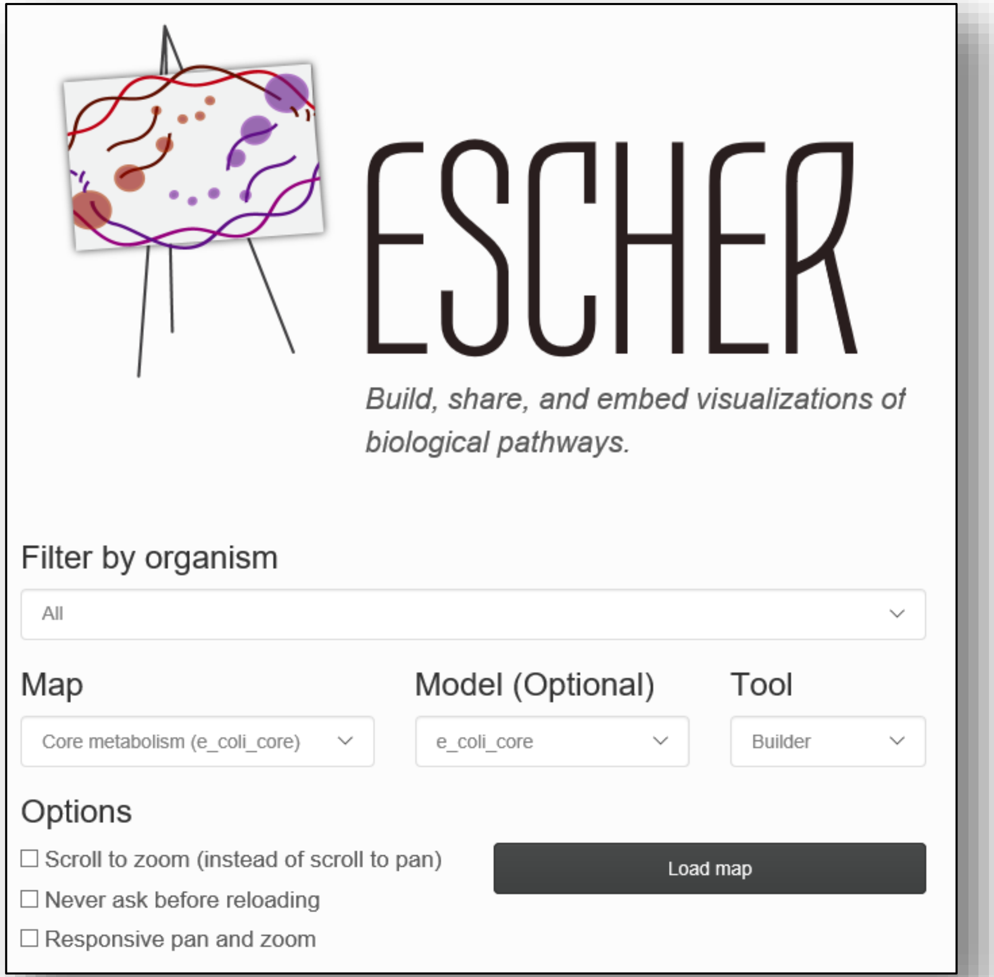
COBRApy Commands for Gene Information

(See *Gene_Overview.ipynb*)

- **Gene Info by Index Number** - model.genes[index]– ex. *model.genes[1]*
- **Gene Info by ID** - model.genes.geneID – ex. *model.metabolites.b1241*
- **Gene Name** - model.genes.geneID.name – ex. *model.metabolites.b1241.name*
- **Gene Functional** - model.genes.geneID.functional – ex. *model.metabolites.b1241.functional*
- **Gene Reactions** - model.genes.geneID.reactions – ex. *model.metabolites.b1241.reactions*

COBRA Models Overview

- COBRApy Toolbox Overview
- COBRA Models Overview
  - ✓ COBRApy Examples
- Reactions
  - ✓ COBRApy Examples
- Metabolites
  - ✓ COBRApy Examples
- Genes
  - ✓ COBRApy Examples
- ➡ Visualization
  - ✓ COBRApy Examples


ESCHER

Build, share, and embed visualizations of biological pathways.

Filter by organism

All

Map: Core metabolism (e_coli_core)

Model (Optional): e_coli_core

Tool: Builder

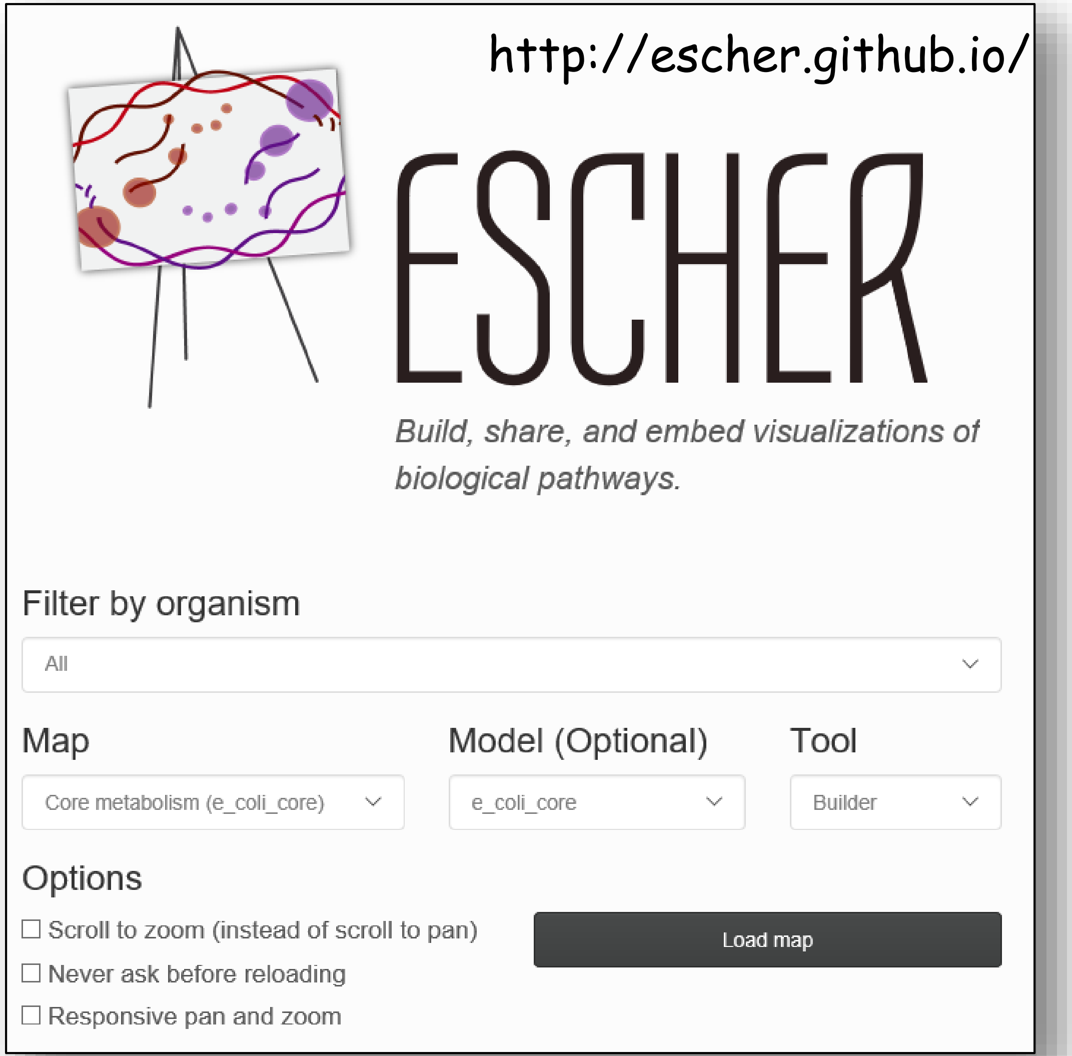
Options

☐ Scroll to zoom (instead of scroll to pan)

☐ Never ask before reloading

☐ Responsive pan and zoom

Escher Visualization



http://escher.github.io/

Build, share, and embed visualizations of biological pathways.

Filter by organism

All

Map: Core metabolism (e_coli_core)

Model (Optional): e_coli_core

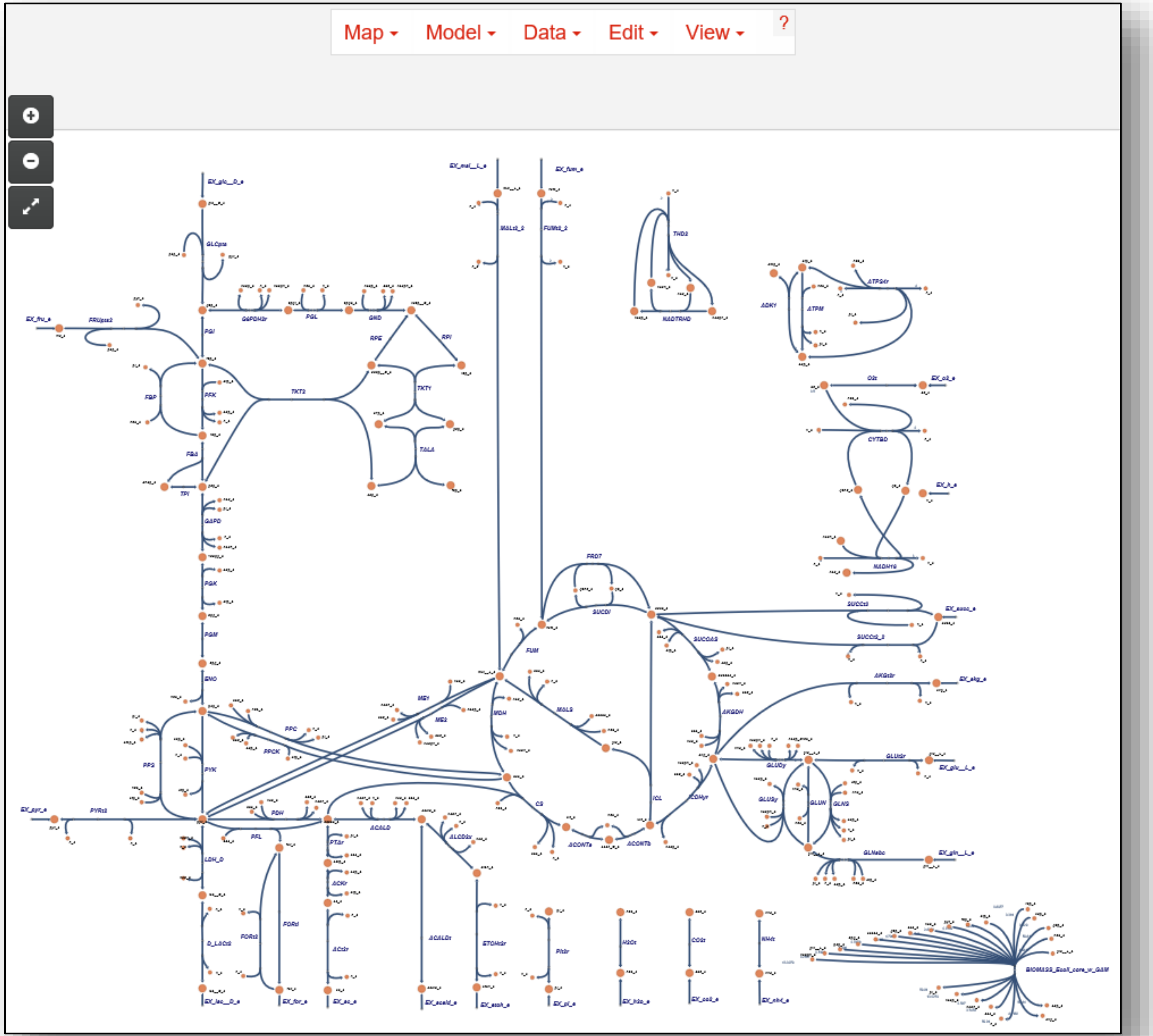
Tool: Builder

Options

☐ Scroll to zoom (instead of scroll to pan)

☐ Never ask before reloading

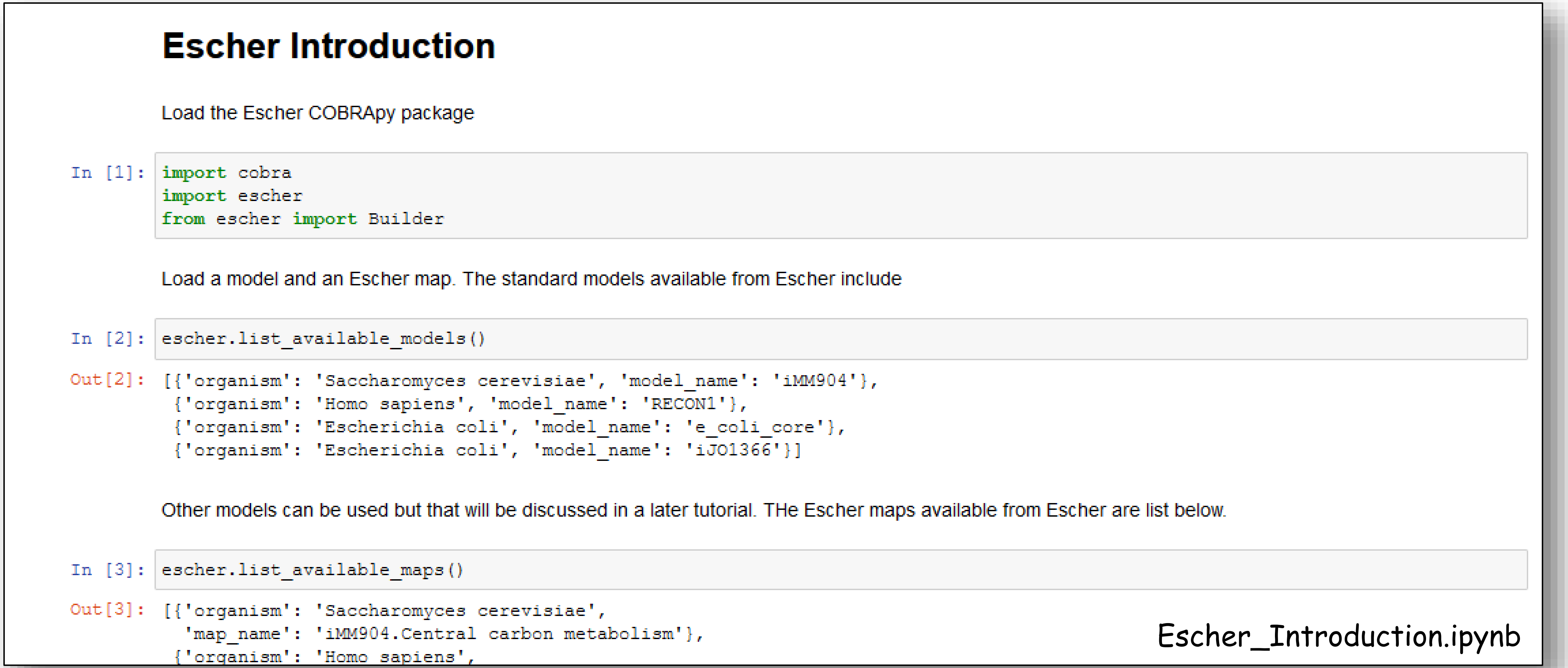
☐ Responsive pan and zoom



Escher Supported Maps and Models

- **Models**
  - 'organism': 'Saccharomyces cerevisiae', 'model_name': 'iMM904'},
  - 'organism': 'Homo sapiens', 'model_name': 'RECON1'},
  - 'organism': 'Escherichia coli', 'model_name': 'e_coli_core'},
  - 'organism': 'Escherichia coli', 'model_name': 'iJO1366'}
- **Maps**
  - {'organism': 'Saccharomyces cerevisiae', 'map_name': 'iMM904.Central carbon metabolism'},
  - {'organism': 'Homo sapiens', 'map_name': 'RECON1.Inositol retinol metabolism'},
  - {'organism': 'Homo sapiens', 'map_name': 'RECON1.Glycolysis TCA PPP'},
  - {'organism': 'Homo sapiens', 'map_name': 'RECON1.Tryptophan metabolism'},
  - {'organism': 'Homo sapiens', 'map_name': 'RECON1.Carbohydrate metabolism'},
  - {'organism': 'Homo sapiens', 'map_name': 'RECON1.Amino acid metabolism (partial)'},
  - {'organism': 'Escherichia coli', 'map_name': 'iJO1366.Nucleotide metabolism'},
  - {'organism': 'Escherichia coli', 'map_name': 'iJO1366.Fatty acid biosynthesis (saturated)'},
  - {'organism': 'Escherichia coli', 'map_name': 'iJO1366.Nucleotide and histidine biosynthesis'},
  - {'organism': 'Escherichia coli', 'map_name': 'e_coli_core.Core metabolism'},
  - {'organism': 'Escherichia coli', 'map_name': 'iJO1366.Central metabolism'},
  - {'organism': 'Escherichia coli', 'map_name': 'iJO1366.Fatty acid beta-oxidation'}]

Escher Introduction



Escher Introduction

Load the Escher COBRApy package

In [1]:
```
import cobra
import escher
from escher import Builder
```

Load a model and an Escher map. The standard models available from Escher include

In [2]:
```
escher.list_available_models()
```

Out[2]:
```
[{'organism': 'Saccharomyces cerevisiae', 'model_name': 'iMM904'},
 {'organism': 'Homo sapiens', 'model_name': 'RECON1'},
 {'organism': 'Escherichia coli', 'model_name': 'e_coli_core'},
 {'organism': 'Escherichia coli', 'model_name': 'iJO1366'}]
```

Other models can be used but that will be discussed in a later tutorial. THe Escher maps available from Escher are list below.

In [3]:
```
escher.list_available_maps()
```

Out[3]:
```
[{'organism': 'Saccharomyces cerevisiae',
  'map_name': 'iMM904.Central carbon metabolism'},
 {'organism': 'Homo sapiens',
```

Escher_Introduction.ipynb

COBRA Models Overview

- COBRA Models Overview
  - ✓ Overview
  - ✓ COBRApy Examples
- Reactions
  - ✓ Overview
  - ✓ COBRApy Examples
- Metabolites
  - ✓ Overview
  - ✓ COBRApy Examples
- Genes
  - ✓ Overview
  - ✓ COBRApy Examples
- Visualization
  - ✓ Overview
  - ✓ COBRApy Examples

The COBRApy Toolbox

Learning Objectives

Each student should be able to:

- Explain the purpose of the COBRApy Toolbox,
- Understand the organization of the COBRA models,
- Demonstrate the ability to interact with COBRA models.

COBRA Models - Reflective Questions

1. Describe the genome-scale metabolic reconstruction process.
2. What is purpose of the BIGG database?
3. What are metabolites in the COBRA models?
4. What are reactions in the COBRA models?
5. What are genes in the COBRA models?
6. What is a pathway in a COBRA model?
7. What is an exchange reaction?
8. What is a transport reaction?
9. Is cellular uptake represented with a positive or negative number?
10. What is a gene-protein-reactions?
11. What are the built-in models available with the standard COBRApy package?
12. What is the purpose of the objective function?
13. What is the purpose of the biomass function?
14. What information is provided for a reaction in COBRApy models?
15. What information is provided for a metabolite in COBRApy models?
16. What information is provided for a gene in COBRApy models?
17. What models are included in the Escher package?
18. What is the difference between an Escher model and an Escher map?

References

1. Thiele, I. and B. O. Palsson (2010). "A protocol for generating a high-quality genome-scale metabolic reconstruction." Nature protocols 5(1): 93-121Insert content here.
2. Schellenberger, J., J. O. Park, et al. (2010). "BiGG: a Biochemical Genetic and Genomic knowledgebase of large scale metabolic reconstructions." BMC Bioinformatics 11: 213.
3. Orth, J. D., I. Thiele, et al. (2010). "What is flux balance analysis?" Nature biotechnology **28(3): 245-248**
4. Becker, S. A., et al. (2007). "Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox." Nature protocols **2(3): 727-738.**
5. Thiele, I. and B. O. Palsson (2010). "A protocol for generating a high-quality genome-scale metabolic reconstruction." Nature protocols 5(1): 93-121.
6. Reconstruction and Use of Microbial Metabolic Networks: the Core Escherichia coli Metabolic Model as an Educational Guide by Orth, Fleming, and Palsson (2010)
7. Ebrahim, A., Lerman, J.A., Palsson, B.O. et al. COBRApy: COnstraints-Based Reconstruction and Analysis for Python. BMC Syst Biol 7, 74 (2013). https://doi.org/10.1186/1752-0509-7-74
8. King, Zachary A., et al. "Escher: a web application for building, sharing, and embedding data-rich visualizations of biological pathways." PLoS computational biology 11.8 (2015): e1004321.