Model Interrogation

Part I

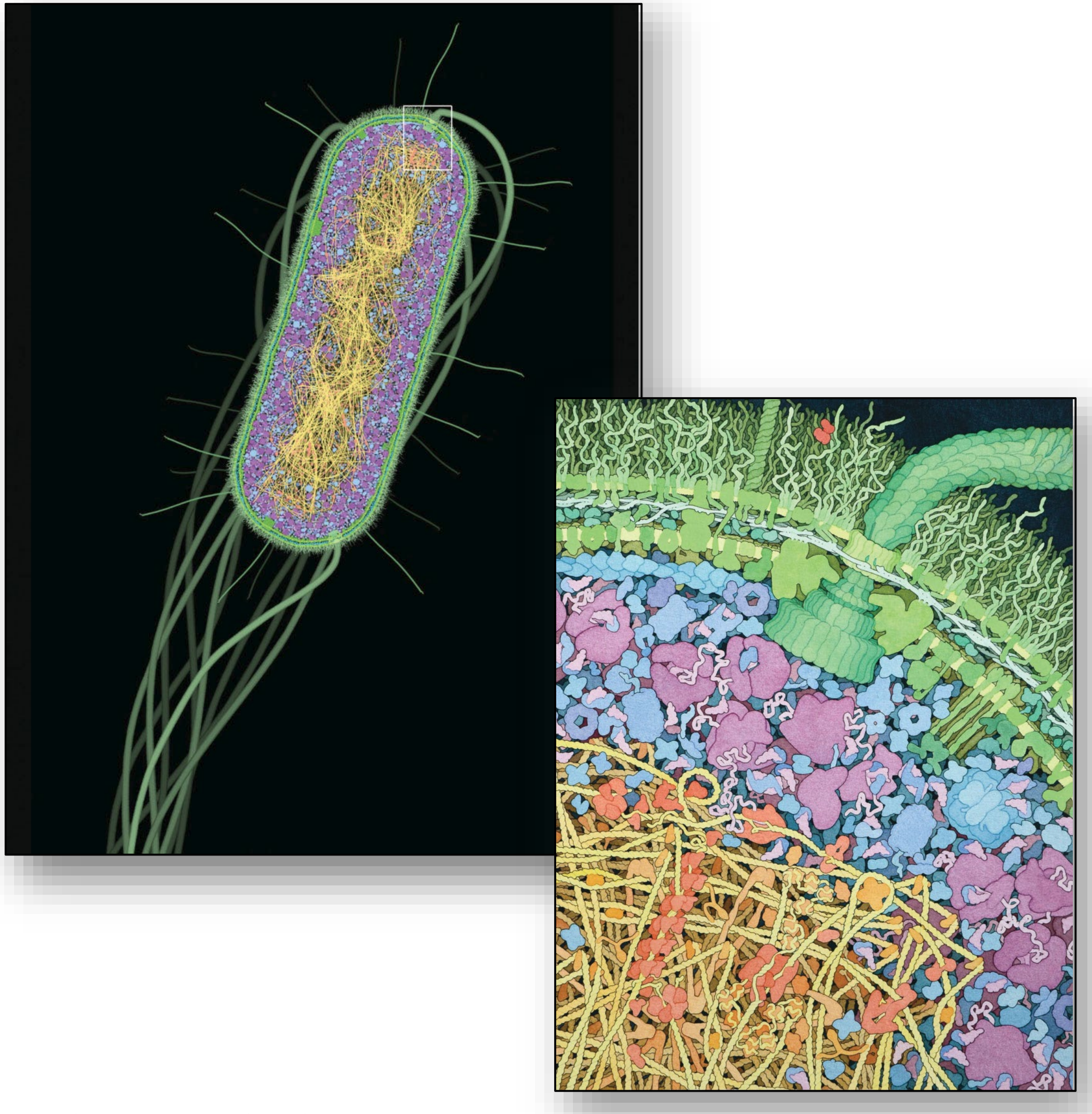
Learning Objectives

Each student should be able to:

- Explain the purpose of model verification,
- Explain the role of model attributes,
- Explain the role of biomass functions,
- Explain the role of carbon/nitrogen/phosphorus/sulfur sources,
- Explain the role of essential genes/reactions,
- Explain the types of *in silico* growth media.

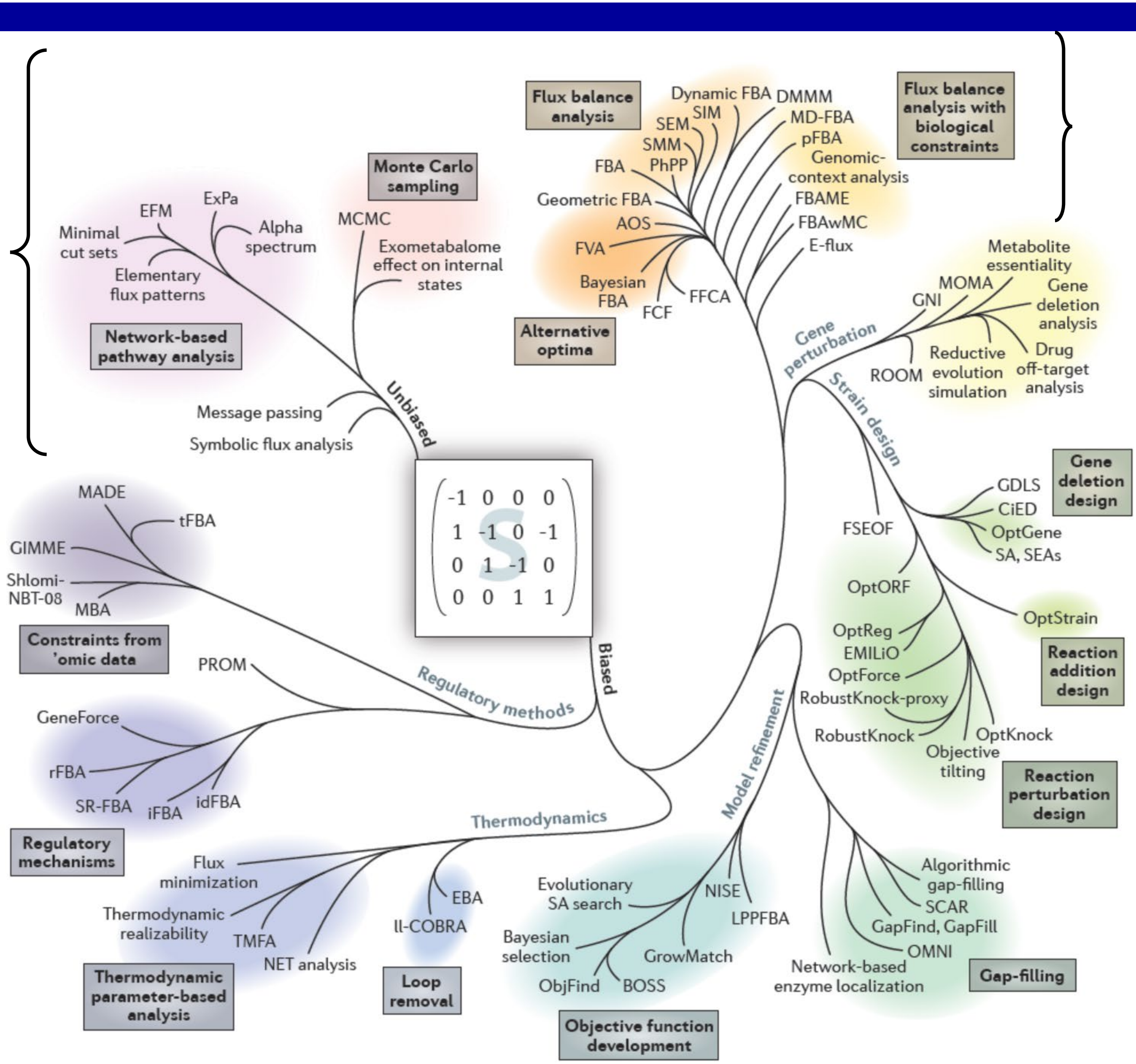
Lesson Outline

- Overview
- Understanding a COBRA Model
  - ✓ Model Verification - Memote
  - ✓ Model Attributes
  - ✓ Biomass Functions
  - ✓ Carbon/Nitrogen/Phosphorus/Sulfur Sources
  - ✓ Essential Genes/Reactions
  - ✓ Growth Media



Goodsell, D. "Miniseries: illustrating the machinery of life." *Escherichia coli. Biochem. Mol. Biol. Educ.* 37 (2009): 325-32.

The 'Phylogeny' of Constraint-based Modeling Methods



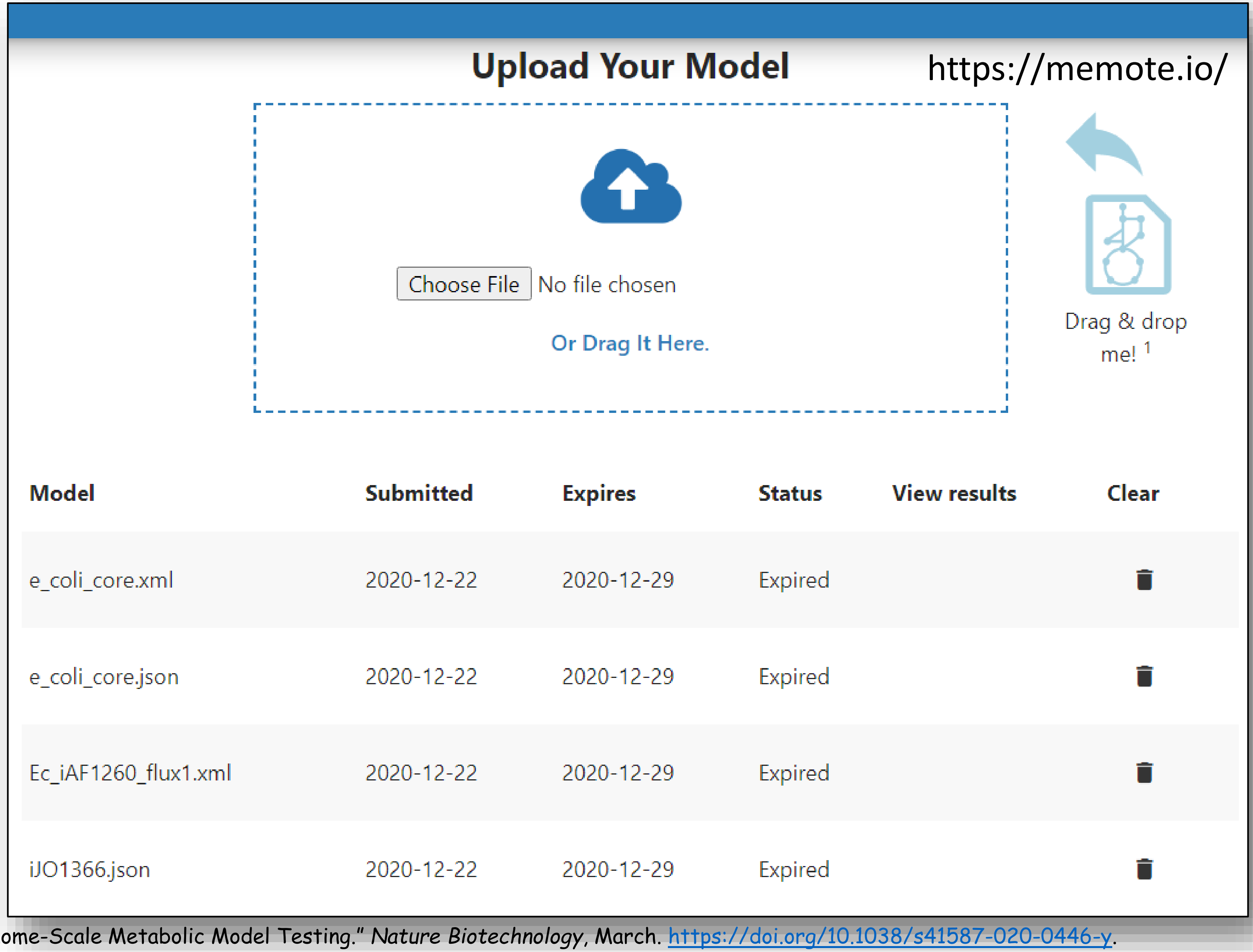
Lewis, N. E., H. Nagarajan, et al. (2012). "Constraining the metabolic genotype-phenotype relationship using a phylogeny of in silico methods." Nature reviews. Microbiology 10(4): 291-305.

Lesson Outline

- Overview
- Understanding a COBRA Model
  - ➡ ✓ Model Verification - Memote
  - ✓ Model Attributes
  - ✓ Biomass Functions
  - ✓ Carbon/Nitrogen/Phosphorus/Sulfur Sources
  - ✓ Essential Genes/Reactions
  - ✓ Growth Media

Memote

- MEMOTE stands for metabolic model testing
- For a given metabolic model, the tests in MEMOTE fulfill two purposes:
  - ✓ They provide an independent, comparable score and a comprehensive overview.
  - ✓ The tests were selected on the basis of previously established guidelines and are continuously curated by the community.
- Its interface is a simple drag & drop of your model of choice.
- MEMOTE accepts models formatted in the current de facto standard, SBML , or a non-standard JSON format.
- Produces an output HTML file that can be downloaded.
- MEMOTE is powered by cobrapy and thus bound to its capabilities.

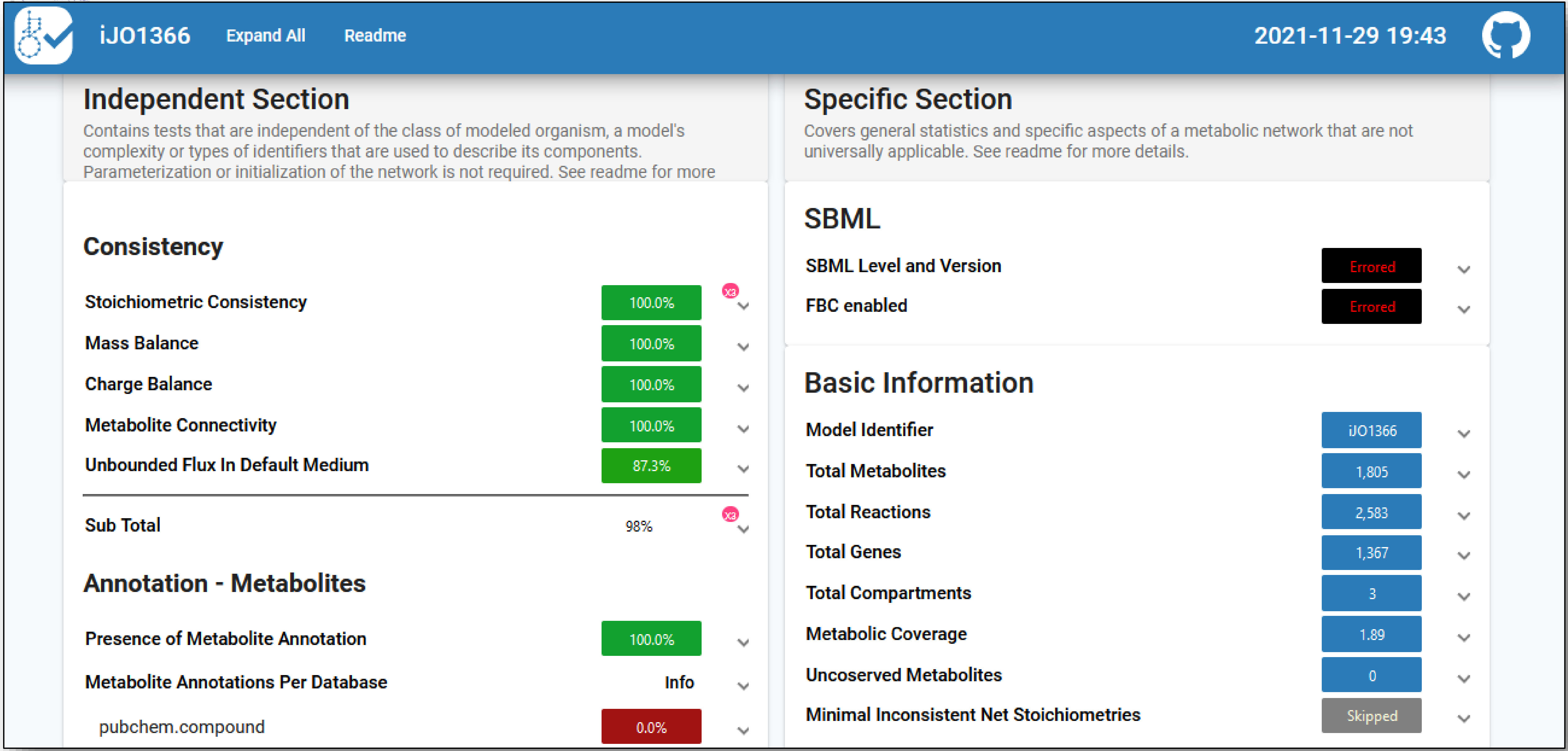


Upload Your Model

https://memote.io/

Model	Submitted	Expires	Status	View results	Clear
e_coli_core.xml	2020-12-22	2020-12-29	Expired		🗑
e_coli_core.json	2020-12-22	2020-12-29	Expired		🗑
Ec_iAF1260_flux1.xml	2020-12-22	2020-12-29	Expired		🗑
iJO1366.json	2020-12-22	2020-12-29	Expired		🗑

Lieven, Christian, et al. 2020. “MEMOTE for Standardized Genome-Scale Metabolic Model Testing.” *Nature Biotechnology*, March. https://doi.org/10.1038/s41587-020-0446-y.



iJO1366 Expand All Readme 2021-11-29 19:43

Independent Section

Contains tests that are independent of the class of modeled organism, a model's complexity or types of identifiers that are used to describe its components. Parameterization or initialization of the network is not required. See readme for more

Consistency

Stoichiometric Consistency	100.0%
Mass Balance	100.0%
Charge Balance	100.0%
Metabolite Connectivity	100.0%
Unbounded Flux In Default Medium	87.3%
Sub Total	98%

Annotation - Metabolites

Presence of Metabolite Annotation	100.0%
Metabolite Annotations Per Database	Info
pubchem.compound	0.0%

Specific Section

Covers general statistics and specific aspects of a metabolic network that are not universally applicable. See readme for more details.

SBML

SBML Level and Version	Errored
FBC enabled	Errored

Basic Information

Model Identifier	iJO1366
Total Metabolites	1,805
Total Reactions	2,583
Total Genes	1,367
Total Compartments	3
Metabolic Coverage	1.89
Uncoserved Metabolites	0
Minimal Inconsistent Net Stoichiometries	Skipped

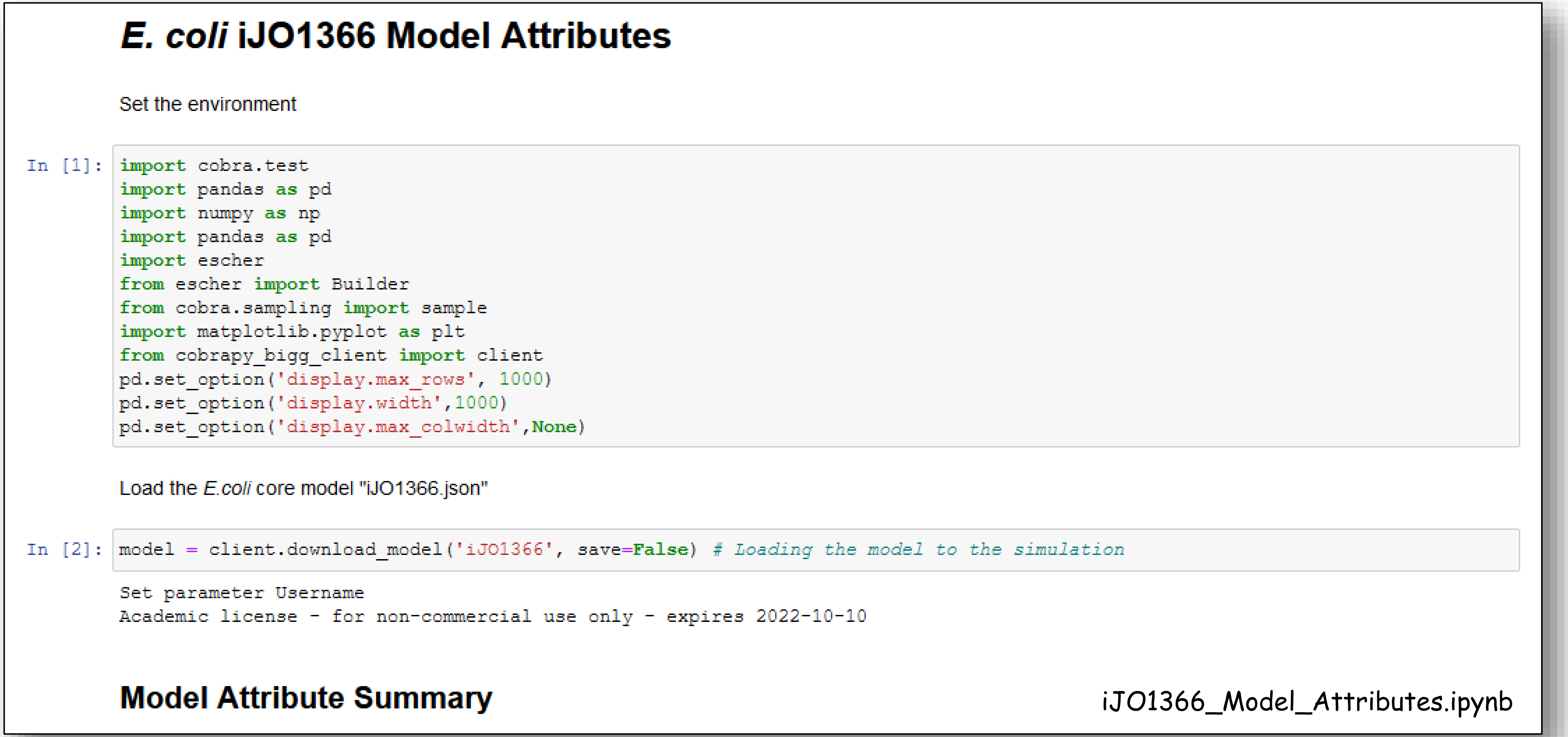
Lesson Outline

- Overview
- Understanding a COBRA Model
  - ✓ Model Verification - Memote
  - ➡ ✓ Model Attributes
  - ✓ Biomass Functions
  - ✓ Carbon/Nitrogen/Phosphorus/Sulfur Sources
  - ✓ Essential Genes/Reactions
  - ✓ Growth Media

Key Model Attributes

1. Model attribute summary
2. Biomass functions
3. Default objective function
4. Compartments
5. Subsystems
6. Reactions (ID, name, formula, lower bound, upper bound, subsystem)
7. Boundary reactions (exchange, demand, sink)
8. Metabolites (ID, name, formula, compartment, charge)
9. Genes (ID, name, gene functional, reactions)
10. Carbon sources
11. Nitrogen sources
12. Phosphorus sources
13. Sulfur sources
14. Default open uptake reactions
15. Default flux values
16. Available Escher maps

Model Attributes



E. coli iJO1366 Model Attributes

Set the environment

```
In [1]: import cobra.test
        import pandas as pd
        import numpy as np
        import pandas as pd
        import escher
        from escher import Builder
        from cobra.sampling import sample
        import matplotlib.pyplot as plt
        from cobrapy_bigg_client import client
        pd.set_option('display.max_rows', 1000)
        pd.set_option('display.width',1000)
        pd.set_option('display.max_colwidth',None)
```

Load the *E.coli* core model "iJO1366.json"

```
In [2]: model = client.download_model('iJO1366', save=False) # Loading the model to the simulation
```

```
Set parameter Username
Academic license - for non-commercial use only - expires 2022-10-10
```

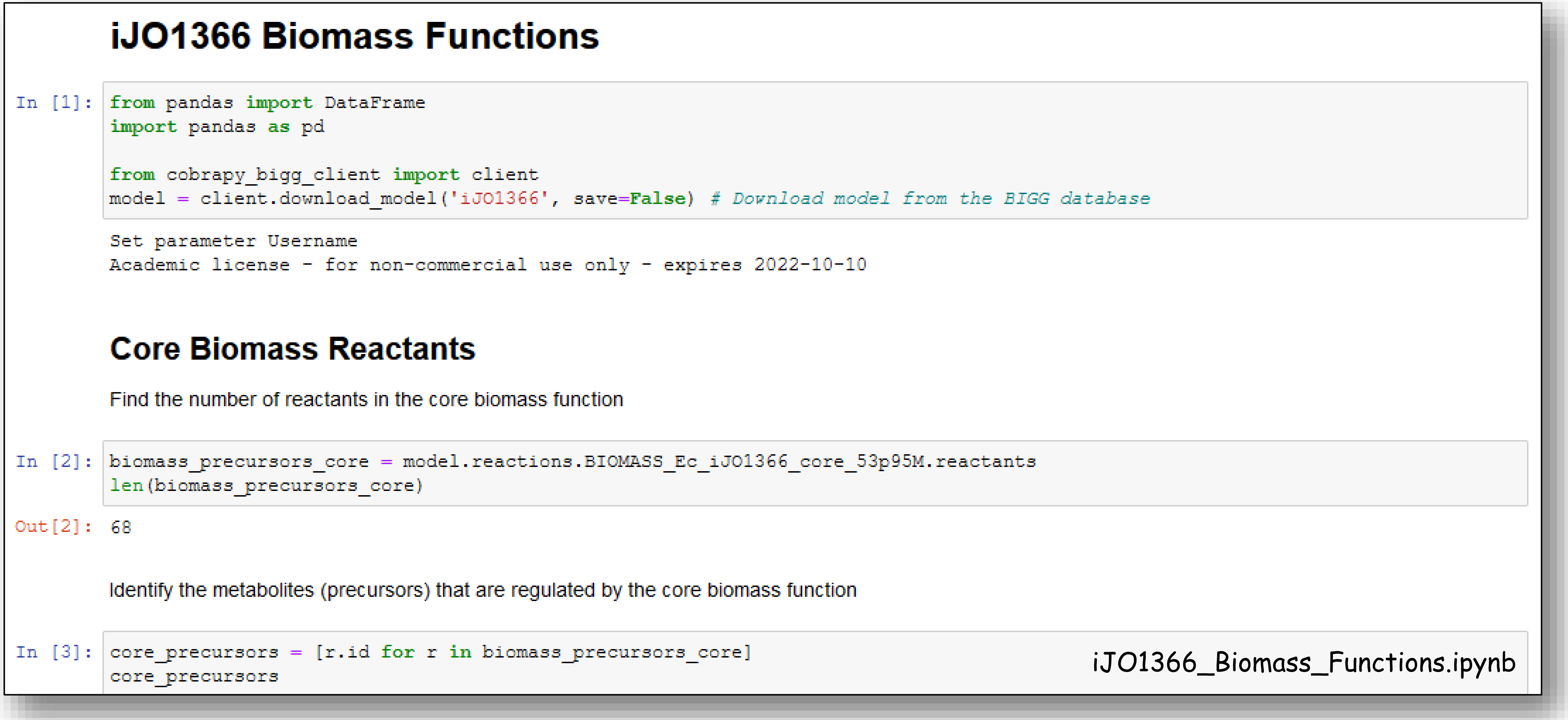
Model Attribute Summary

iJO1366_Model_Attributes.ipynb

Lesson Outline

- Overview
- Understanding a COBRA Model
  - ✓ Model Verification - Memote
  - ✓ Model Attributes
  - ➡ ✓ Biomass Functions
  - ✓ Carbon/Nitrogen/Phosphorus/Sulfur Sources
  - ✓ Essential Genes/Reactions
  - ✓ Growth Media

Biomass Functions



iJO1366 Biomass Functions

In [1]:
```
from pandas import DataFrame
import pandas as pd

from cobrapy_bigg_client import client
model = client.download_model('iJO1366', save=False) # Download model from the BIGG database
```

```
Set parameter Username
Academic license - for non-commercial use only - expires 2022-10-10
```

Core Biomass Reactants

Find the number of reactants in the core biomass function

In [2]:
```
biomass_precursors_core = model.reactions.BIOMASS_Ec_iJO1366_core_53p95M.reactants
len(biomass_precursors_core)
```

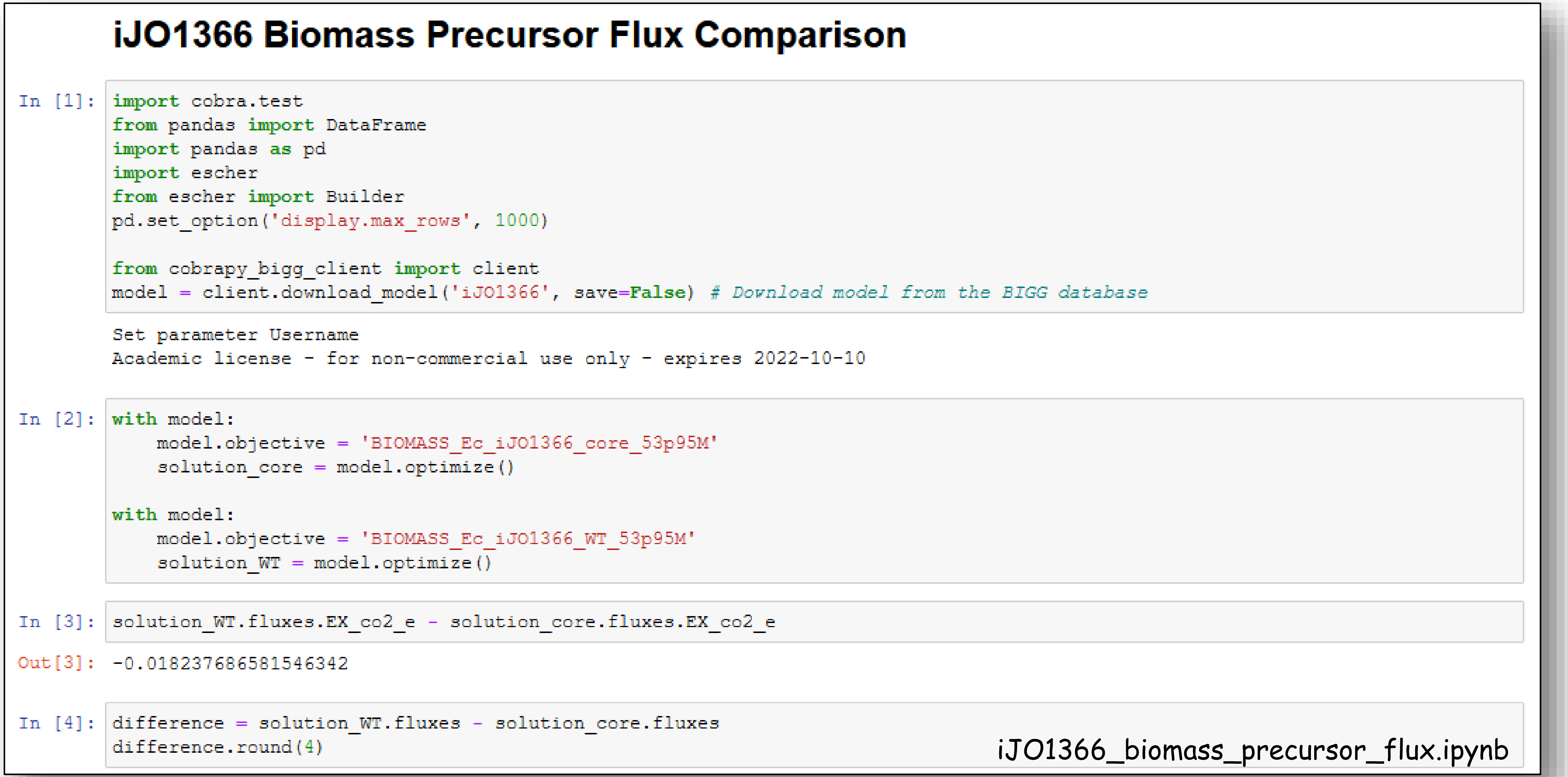
Out[2]:
```
68
```

Identify the metabolites (precursors) that are regulated by the core biomass function

In [3]:
```
core_precursors = [r.id for r in biomass_precursors_core]
core_precursors
```

iJO1366_Biomass_Functions.ipynb

Biomass Flux



iJO1366 Biomass Precursor Flux Comparison

```
In [1]: import cobra.test
        from pandas import DataFrame
        import pandas as pd
        import escher
        from escher import Builder
        pd.set_option('display.max_rows', 1000)

        from cobrapy_bigg_client import client
        model = client.download_model('iJO1366', save=False) # Download model from the BIGG database

        Set parameter Username
        Academic license - for non-commercial use only - expires 2022-10-10
```

```
In [2]: with model:
            model.objective = 'BIOMASS_Ec_iJO1366_core_53p95M'
            solution_core = model.optimize()

        with model:
            model.objective = 'BIOMASS_Ec_iJO1366_WT_53p95M'
            solution_WT = model.optimize()
```

```
In [3]: solution_WT.fluxes.EX_co2_e - solution_core.fluxes.EX_co2_e

Out[3]: -0.018237686581546342
```

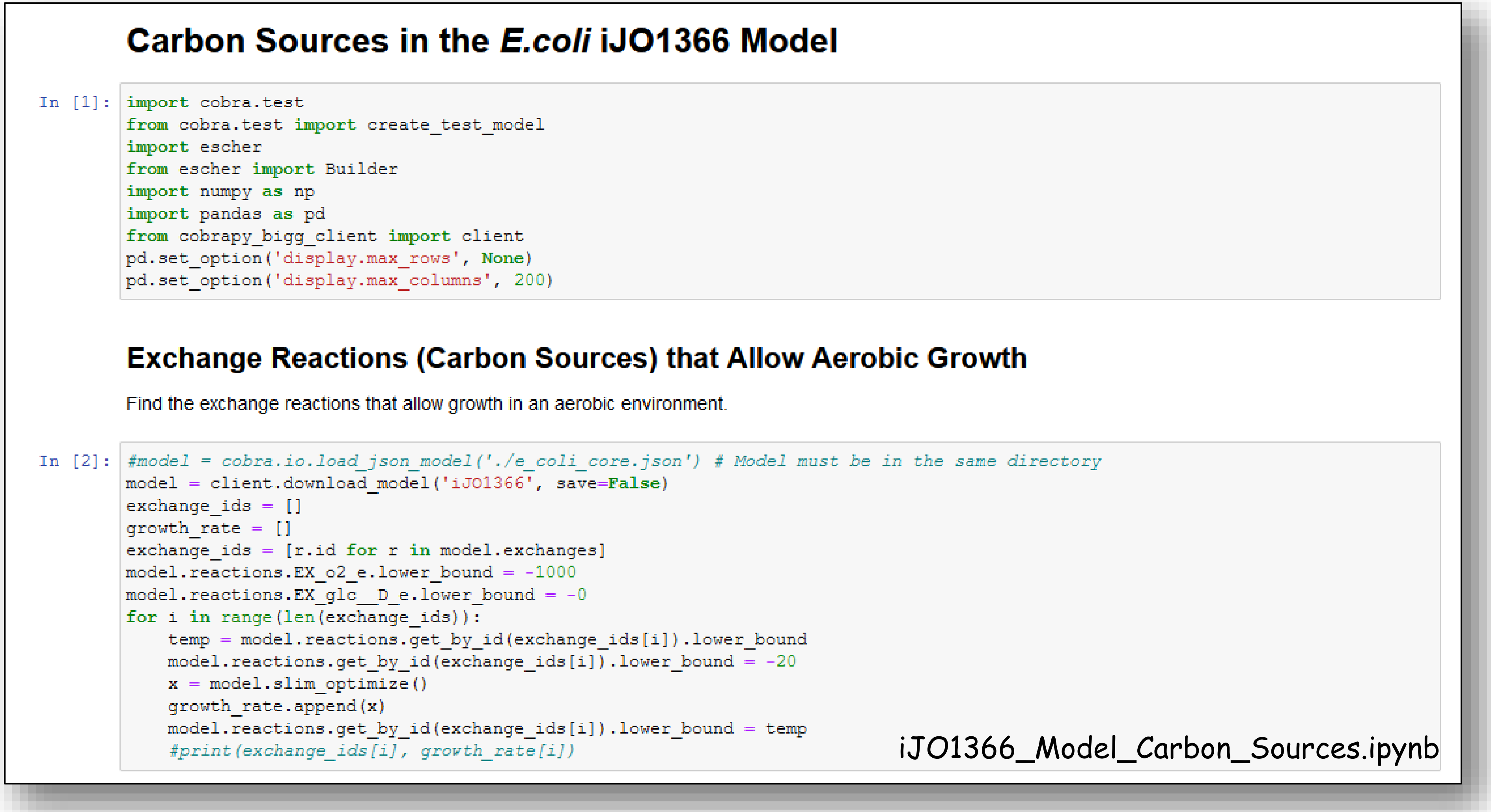
```
In [4]: difference = solution_WT.fluxes - solution_core.fluxes
        difference.round(4)
```

iJO1366_biomass_precursor_flux.ipynb

Lesson Outline

- Overview
- Understanding a COBRA Model
  - ✓ Model Verification - Memote
  - ✓ Model Attributes
  - ✓ Biomass Functions
  - ➡ ✓ Carbon/Nitrogen/Phosphorus/Sulfur Sources
  - ✓ Essential Genes/Reactions
  - ✓ Growth Media

Carbon Sources



Carbon Sources in the *E.coli* iJO1366 Model

In [1]:

```python
import cobra.test
from cobra.test import create_test_model
import escher
from escher import Builder
import numpy as np
import pandas as pd
from cobrapy_bigg_client import client
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', 200)
```

Exchange Reactions (Carbon Sources) that Allow Aerobic Growth

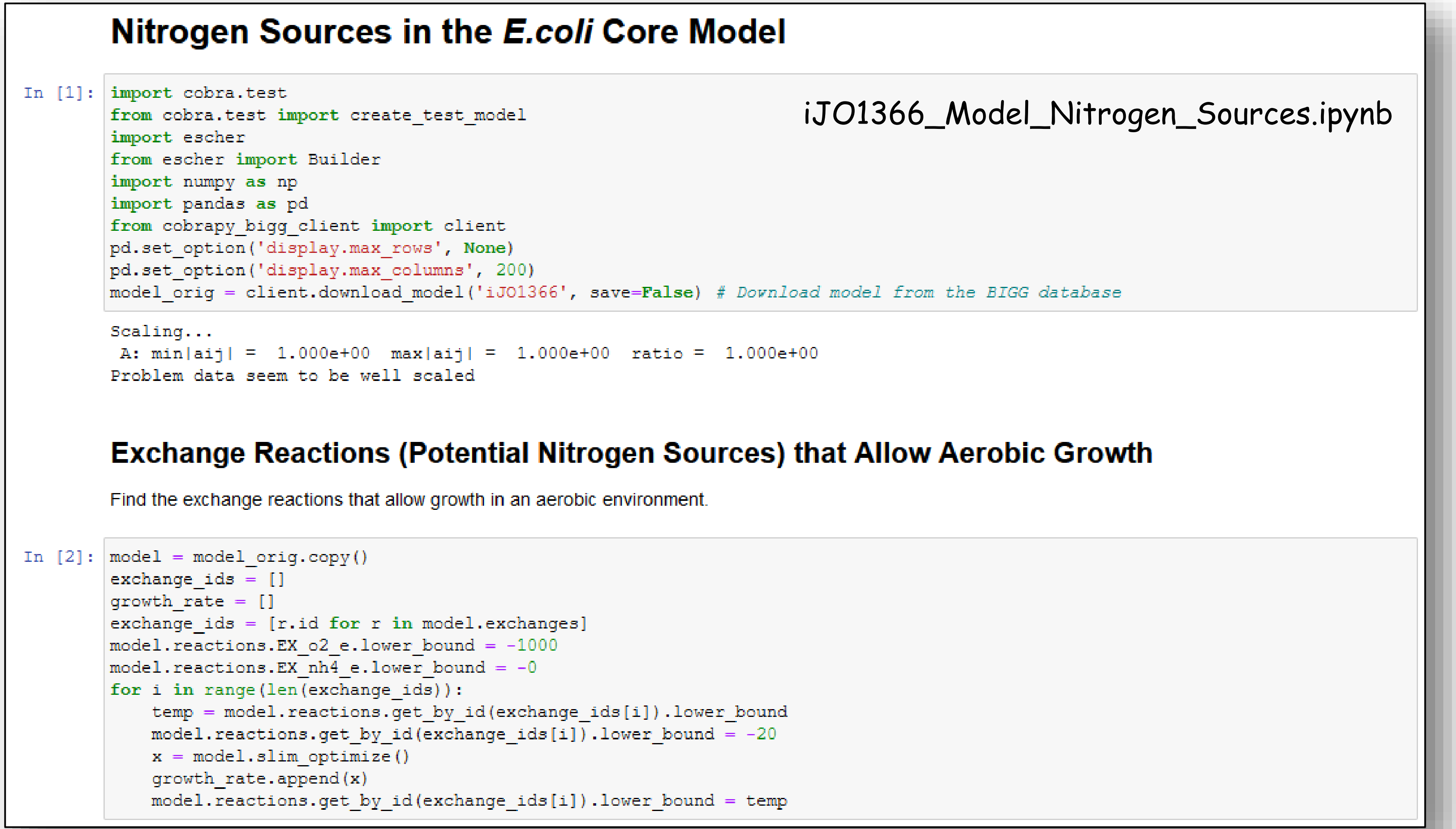
Find the exchange reactions that allow growth in an aerobic environment.

In [2]:

```python
#model = cobra.io.load_json_model('./e_coli_core.json') # Model must be in the same directory
model = client.download_model('iJO1366', save=False)
exchange_ids = []
growth_rate = []
exchange_ids = [r.id for r in model.exchanges]
model.reactions.EX_o2_e.lower_bound = -1000
model.reactions.EX_glc__D_e.lower_bound = -0
for i in range(len(exchange_ids)):
    temp = model.reactions.get_by_id(exchange_ids[i]).lower_bound
    model.reactions.get_by_id(exchange_ids[i]).lower_bound = -20
    x = model.slim_optimize()
    growth_rate.append(x)
    model.reactions.get_by_id(exchange_ids[i]).lower_bound = temp
    #print(exchange_ids[i], growth_rate[i])
```

iJO1366_Model_Carbon_Sources.ipynb

Nitrogen Sources



Nitrogen Sources in the *E.coli* Core Model

iJO1366_Model_Nitrogen_Sources.ipynb

```
In [1]: import cobra.test
        from cobra.test import create_test_model
        import escher
        from escher import Builder
        import numpy as np
        import pandas as pd
        from cobrapy_bigg_client import client
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', 200)
        model_orig = client.download_model('iJO1366', save=False) # Download model from the BIGG database
```

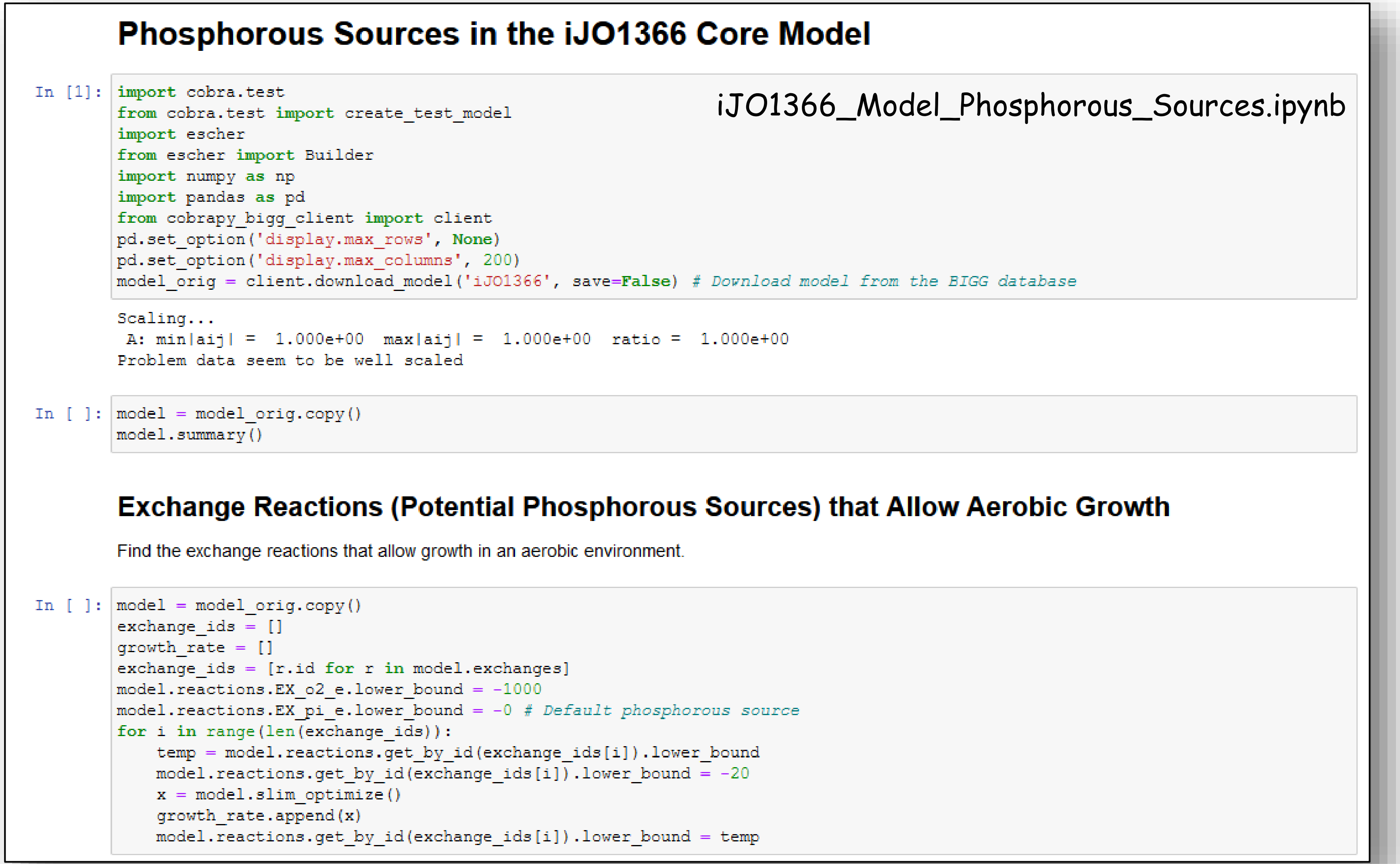
```
Scaling...
 A: min|aij| =  1.000e+00  max|aij| =  1.000e+00  ratio =  1.000e+00
Problem data seem to be well scaled
```

Exchange Reactions (Potential Nitrogen Sources) that Allow Aerobic Growth

Find the exchange reactions that allow growth in an aerobic environment.

```
In [2]: model = model_orig.copy()
        exchange_ids = []
        growth_rate = []
        exchange_ids = [r.id for r in model.exchanges]
        model.reactions.EX_o2_e.lower_bound = -1000
        model.reactions.EX_nh4_e.lower_bound = -0
        for i in range(len(exchange_ids)):
            temp = model.reactions.get_by_id(exchange_ids[i]).lower_bound
            model.reactions.get_by_id(exchange_ids[i]).lower_bound = -20
            x = model.slim_optimize()
            growth_rate.append(x)
            model.reactions.get_by_id(exchange_ids[i]).lower_bound = temp
```

Phosphorus Sources



iJO1366_Model_Phosphorous_Sources.ipynb

Phosphorous Sources in the iJO1366 Core Model

```
In [1]: import cobra.test
        from cobra.test import create_test_model
        import escher
        from escher import Builder
        import numpy as np
        import pandas as pd
        from cobrapy_bigg_client import client
        pd.set_option('display.max_rows', None)
        pd.set_option('display.max_columns', 200)
        model_orig = client.download_model('iJO1366', save=False) # Download model from the BIGG database
```

```
Scaling...
 A: min|aij| =  1.000e+00  max|aij| =  1.000e+00  ratio =  1.000e+00
Problem data seem to be well scaled
```

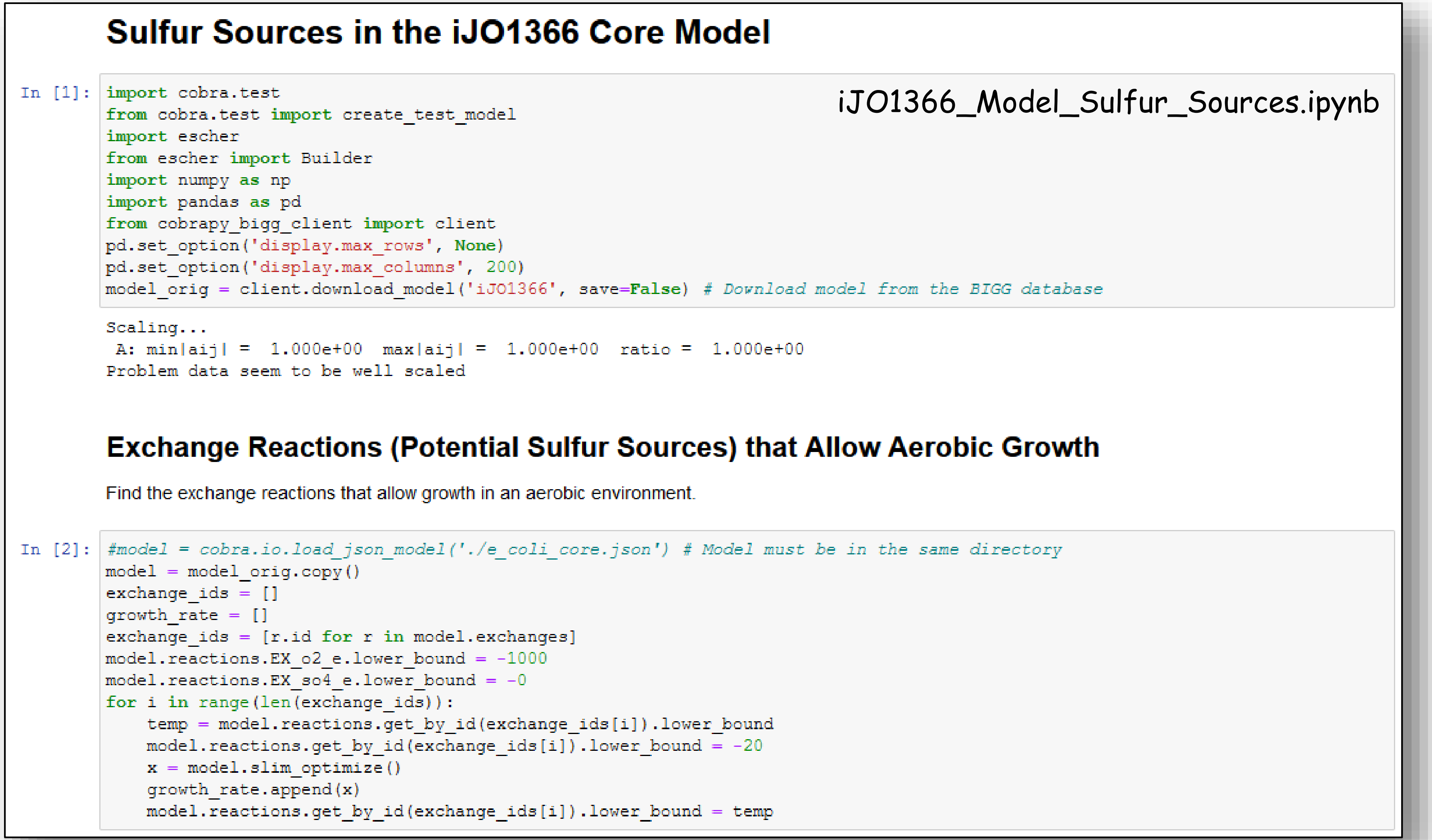
```
In [ ]: model = model_orig.copy()
        model.summary()
```

Exchange Reactions (Potential Phosphorous Sources) that Allow Aerobic Growth

Find the exchange reactions that allow growth in an aerobic environment.

```
In [ ]: model = model_orig.copy()
        exchange_ids = []
        growth_rate = []
        exchange_ids = [r.id for r in model.exchanges]
        model.reactions.EX_o2_e.lower_bound = -1000
        model.reactions.EX_pi_e.lower_bound = -0 # Default phosphorous source
        for i in range(len(exchange_ids)):
            temp = model.reactions.get_by_id(exchange_ids[i]).lower_bound
            model.reactions.get_by_id(exchange_ids[i]).lower_bound = -20
            x = model.slim_optimize()
            growth_rate.append(x)
            model.reactions.get_by_id(exchange_ids[i]).lower_bound = temp
```

Sulfur Sources



Sulfur Sources in the iJO1366 Core Model

iJO1366_Model_Sulfur_Sources.ipynb

In [1]:
```
import cobra.test
from cobra.test import create_test_model
import escher
from escher import Builder
import numpy as np
import pandas as pd
from cobrapy_bigg_client import client
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', 200)
model_orig = client.download_model('iJO1366', save=False) # Download model from the BIGG database
```

```
Scaling...
 A: min|aij| =  1.000e+00  max|aij| =  1.000e+00  ratio =  1.000e+00
Problem data seem to be well scaled
```

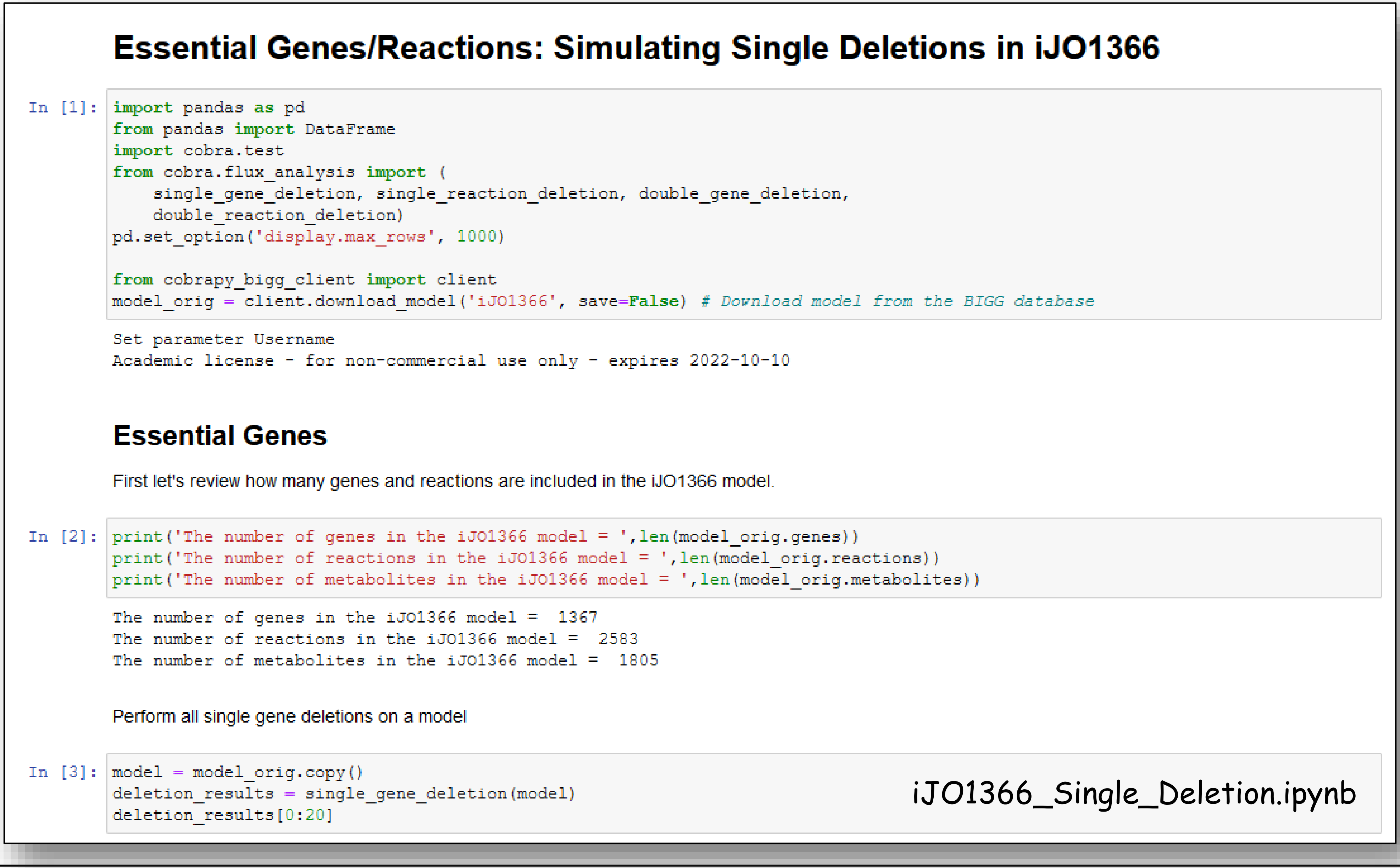
Exchange Reactions (Potential Sulfur Sources) that Allow Aerobic Growth

Find the exchange reactions that allow growth in an aerobic environment.

In [2]:
```
#model = cobra.io.load_json_model('./e_coli_core.json') # Model must be in the same directory
model = model_orig.copy()
exchange_ids = []
growth_rate = []
exchange_ids = [r.id for r in model.exchanges]
model.reactions.EX_o2_e.lower_bound = -1000
model.reactions.EX_so4_e.lower_bound = -0
for i in range(len(exchange_ids)):
    temp = model.reactions.get_by_id(exchange_ids[i]).lower_bound
    model.reactions.get_by_id(exchange_ids[i]).lower_bound = -20
    x = model.slim_optimize()
    growth_rate.append(x)
    model.reactions.get_by_id(exchange_ids[i]).lower_bound = temp
```

Lesson Outline

- Overview
- Understanding a COBRA Model
    - ✓ Model Verification - Memote
    - ✓ Model Attributes
    - ✓ Biomass Functions
    - ✓ Carbon/Nitrogen/Phosphorus/Sulfur Sources
    - ➡ ✓ Essential Genes/Reactions
    - ✓ Growth Media



Essential Genes/Reactions: Simulating Single Deletions in iJO1366

```
In [1]: import pandas as pd
        from pandas import DataFrame
        import cobra.test
        from cobra.flux_analysis import (
            single_gene_deletion, single_reaction_deletion, double_gene_deletion,
            double_reaction_deletion)
        pd.set_option('display.max_rows', 1000)

        from cobrapy_bigg_client import client
        model_orig = client.download_model('iJO1366', save=False) # Download model from the BIGG database
```

```
Set parameter Username
Academic license - for non-commercial use only - expires 2022-10-10
```

Essential Genes

First let's review how many genes and reactions are included in the iJO1366 model.

```
In [2]: print('The number of genes in the iJO1366 model = ',len(model_orig.genes))
        print('The number of reactions in the iJO1366 model = ',len(model_orig.reactions))
        print('The number of metabolites in the iJO1366 model = ',len(model_orig.metabolites))
```

```
The number of genes in the iJO1366 model =  1367
The number of reactions in the iJO1366 model =  2583
The number of metabolites in the iJO1366 model =  1805
```

Perform all single gene deletions on a model

```
In [3]: model = model_orig.copy()
        deletion_results = single_gene_deletion(model)
        deletion_results[0:20]
```

iJO1366_Single_Deletion.ipynb

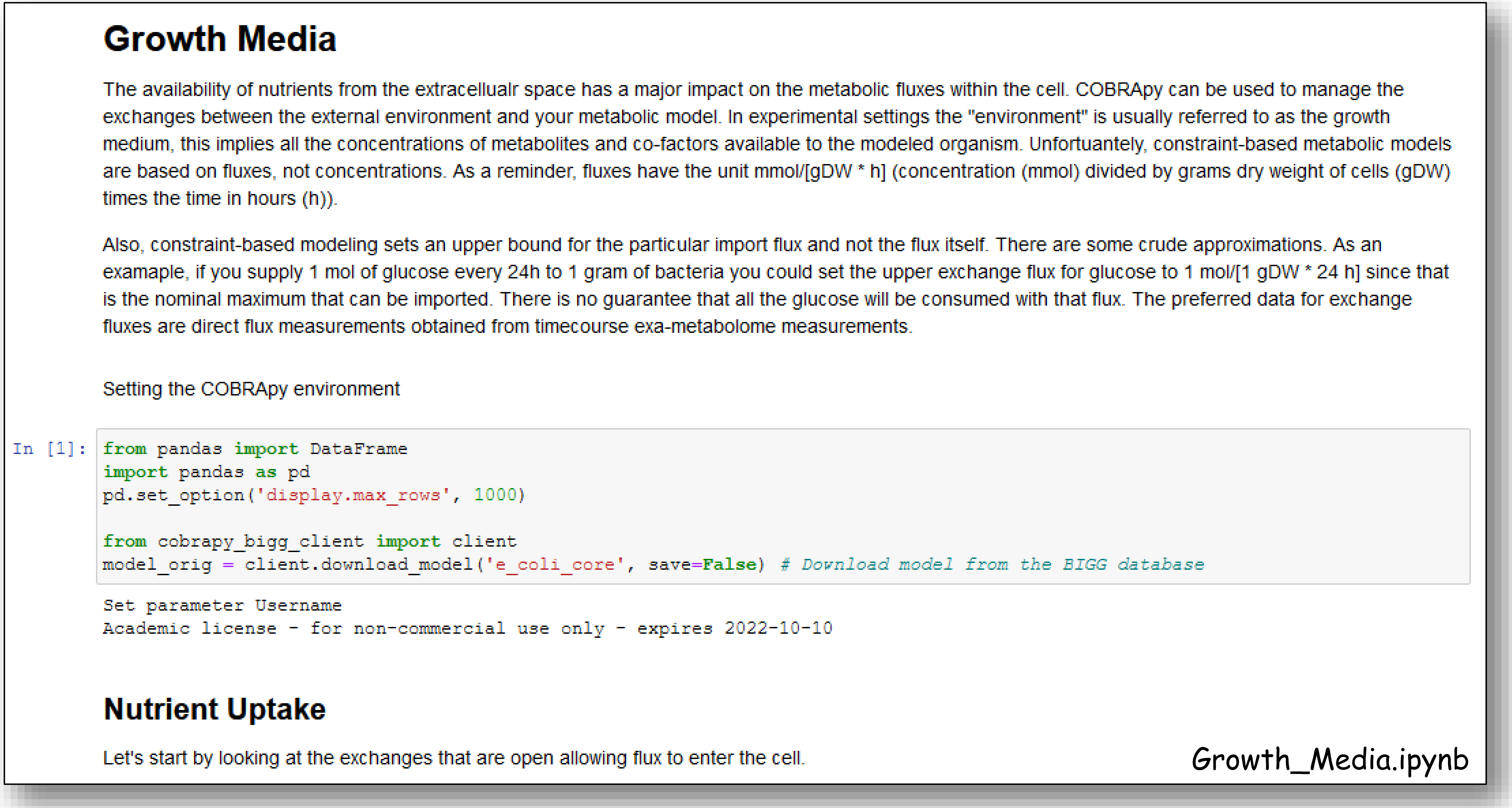
Lesson Outline

- Overview
- Understanding a COBRA Model
  - ✓ Model Verification - Memote
  - ✓ Model Attributes
  - ✓ Biomass Functions
  - ✓ Carbon/Nitrogen/Phosphorus/Sulfur Sources
  - ✓ Essential Genes/Reactions
  - ➡ ✓ Growth Media

Media Comparison

- *Minimal media* are those that contain the minimum nutrients possible for colony growth, generally without the presence of amino acids, and are often used to grow "wild type" microorganisms. Minimal medium typically contains: a carbon source for bacterial growth, which may be a sugar such as glucose, water, and various salts which may vary among bacteria species and growing conditions; these generally provide essential elements such as magnesium, nitrogen, phosphorus, and sulfur to allow the bacteria to synthesize protein and nucleic acids.
- *LB (Lysogeny Broth)* is the most commonly used medium for culturing *E. coli*. It is easy to make, it has rich nutrient contents and its osmolarity is optimal for growth at early log phase. All these features make it adequate for protein production and compensate for the fact that it is not the best option for achieving high cell density cultures. Despite being a rich broth, cell growth stops at a relatively low density.
- An *undefined medium* (also known as a basal or complex medium) includes a carbon source such as glucose for bacterial growth, water, various salts needed for bacterial growth, a source of amino acids and nitrogen (e.g., beef, yeast extract) . This is an undefined medium because the amino acid source contains a variety of compounds with the exact composition being unknown.
- In autoinduction media, a mixture of glucose, lactose, and glycerol is used in an optimized blend. Glucose is the preferred carbon source and is metabolized preferentially during growth, which prevents uptake of lactose until glucose is depleted, usually in mid to late log phase. Consumption of glycerol and lactose follows, the latter being also the inducer of lac-controlled protein expression. In this way, biomass monitoring for timely inducer addition is avoided, as well as culture manipulation

Rosado, G. L. and E. A. Ceccarelli (2014). "Recombinant protein expression in Escherichia coli: advances and challenges." Frontiers in microbiology 5: 172.



Growth Media

The availability of nutrients from the extracellualr space has a major impact on the metabolic fluxes within the cell. COBRApy can be used to manage the exchanges between the external environment and your metabolic model. In experimental settings the "environment" is usually referred to as the growth medium, this implies all the concentrations of metabolites and co-factors available to the modeled organism. Unfortuantely, constraint-based metabolic models are based on fluxes, not concentrations. As a reminder, fluxes have the unit mmol/[gDW * h] (concentration (mmol) divided by grams dry weight of cells (gDW) times the time in hours (h)).

Also, constraint-based modeling sets an upper bound for the particular import flux and not the flux itself. There are some crude approximations. As an examaple, if you supply 1 mol of glucose every 24h to 1 gram of bacteria you could set the upper exchange flux for glucose to 1 mol/[1 gDW * 24 h] since that is the nominal maximum that can be imported. There is no guarantee that all the glucose will be consumed with that flux. The preferred data for exchange fluxes are direct flux measurements obtained from timecourse exa-metabolome measurements.

Setting the COBRApy environment

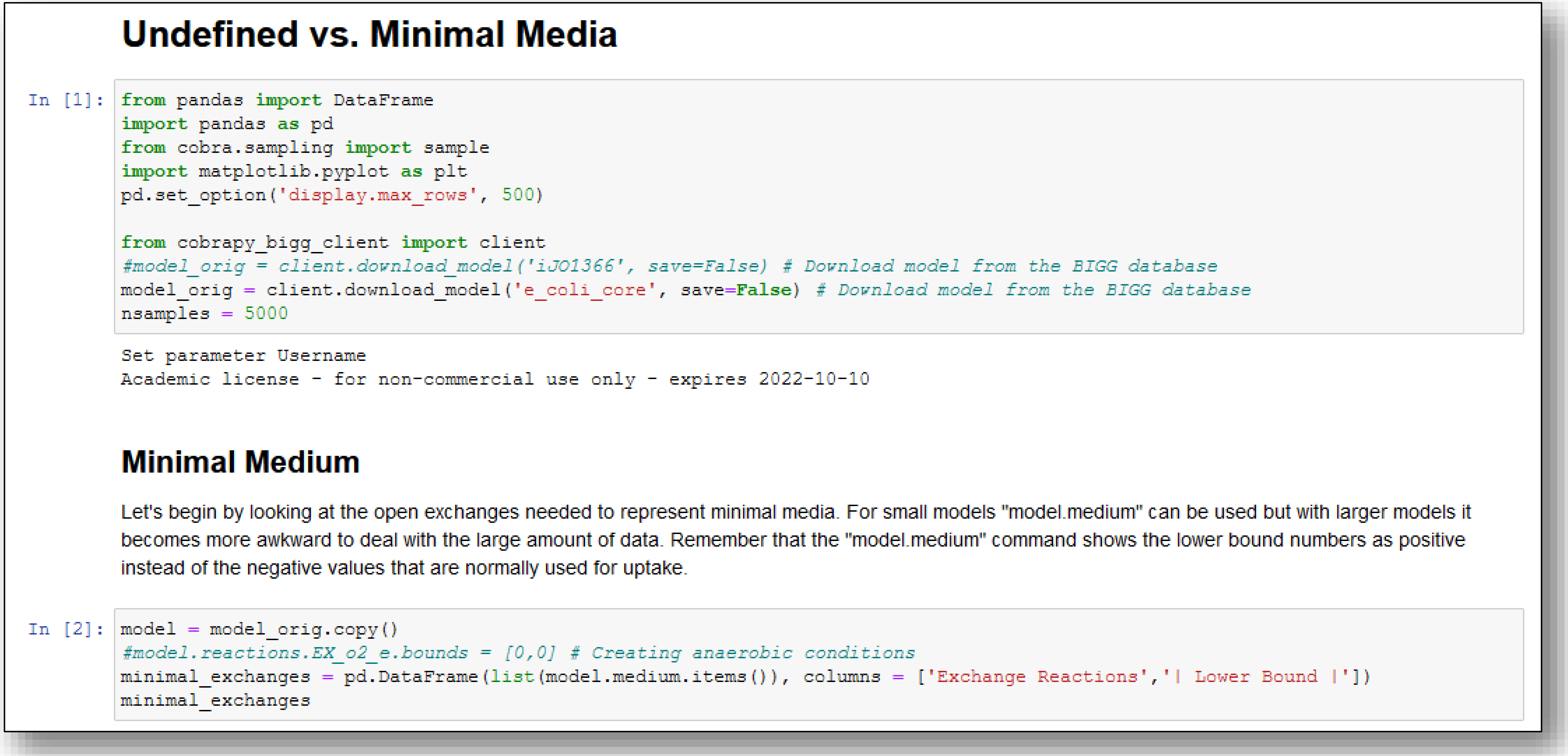
```
In [1]: from pandas import DataFrame
        import pandas as pd
        pd.set_option('display.max_rows', 1000)

        from cobrapy_bigg_client import client
        model_orig = client.download_model('e_coli_core', save=False) # Download model from the BIGG database
```

```
Set parameter Username
Academic license - for non-commercial use only - expires 2022-10-10
```

Nutrient Uptake

Let's start by looking at the exchanges that are open allowing flux to enter the cell.

Growth_Media.ipynb



Undefined vs. Minimal Media

```
In [1]: from pandas import DataFrame
        import pandas as pd
        from cobra.sampling import sample
        import matplotlib.pyplot as plt
        pd.set_option('display.max_rows', 500)

        from cobrapy_bigg_client import client
        #model_orig = client.download_model('iJO1366', save=False) # Download model from the BIGG database
        model_orig = client.download_model('e_coli_core', save=False) # Download model from the BIGG database
        nsamples = 5000
```

```
Set parameter Username
Academic license - for non-commercial use only - expires 2022-10-10
```

Minimal Medium

Let's begin by looking at the open exchanges needed to represent minimal media. For small models "model.medium" can be used but with larger models it becomes more awkward to deal with the large amount of data. Remember that the "model.medium" command shows the lower bound numbers as positive instead of the negative values that are normally used for uptake.

```
In [2]: model = model_orig.copy()
        #model.reactions.EX_o2_e.bounds = [0,0] # Creating anaerobic conditions
        minimal_exchanges = pd.DataFrame(list(model.medium.items()), columns = ['Exchange Reactions','| Lower Bound |'])
        minimal_exchanges
```

Undefined_Medium.ipynb

Lesson Outline

- Overview
- Understanding a COBRA Model
  - ✓ Model Verification - Memote
  - ✓ Model Attributes
  - ✓ Biomass Functions
  - ✓ Carbon/Nitrogen/Phosphorus/Sulfur Sources
  - ✓ Essential Genes/Reactions
  - ✓ Growth Media

Reflective Questions

1. What is the purpose of Memote?
2. Why is model verification important?
3. What does the model's "carbon sources" mean?
4. What does the model's "nitrogen sources" mean?
5. What does the model's "phosphorus sources" mean?
6. What does the model's "sulfur sources" mean?
7. Can a model have more than one biomass function?
8. What is an essential gene?
9. What is an essential reaction?
10. What is a minimal media?
11. What is an undefined media?

M9 Minimal Medium

- One liter of M9 medium (Sigma catalog no. 6030) contains:
  - ✓ $Na_2HPO_4 \cdot 7H_2O$ (6.8g), KH_2PO_4 (3g), NaCl (0.5g), NH_4Cl (1g), $MgSO_4$ (2 mM), $CaCl_2$ (0.1 mM)
- Growth on minimal medium was simulated by maximizing flux through a defined biomass objective function and allowing the uptake of
  - ✓ NH_4, SO_4, O_2, and P_i and the free exchange of H^+, H_2O, and CO_2
- All exchange reaction lower constraints, except the following, should be greater than zero
  - ✓ $-1000 \leq NH_4, SO_4, O_2$, and $P_i \leq 0$
  - ✓ $-1000 \leq H^+, H_2O$, and $CO_2 \leq 1000$
  - ✓ $-1000 \leq$ Carbon source ≤ 0

Recipe for M-9 Minimal Media

- 5X M9 basis
  - $Na_2HPO_4.12\ H_2O$ 85.7 g
  - KH_2PO_4 15.0 g
  - NaCl 2.5 g
  - Dissolve above components in 1000 ml of milli-Q and autoclave
- 5 g $(NH_4)2SO_4$ in 15 ml of H2O
- Trace elements
  - 1 g EDTA
  - 29 mg $ZnSO_4.7H_2O$
  - 198 mg $MnCl_2.\ 4H_2O$
  - 254 mg $CoCl_2.\ 6H_2O$
  - 13.4 mg $CuCl_2$
  - 147 mg $CaCl_2$
  - Dissolve in 100 ml of milli-Q and autoclave
- 20% (w/v) glucose: 25 g in 100 ml of milliQ and filter with 0.22 micron filter
- 0.1 M $CaCl_2.2H_2O$: 1.47 g in 100 ml milliQ and filter with 0.22 micron filter
- 1M $MgSO_4.7H_2O$: 24. 65 g in 100 ml milliQ and filter with 0.22 micron filter
- 10 mM $FeSO_4.7H_2O$: 140 mg in 50 ml of milliQ (prepare fresh)
- 1% thiamine: 500mg in 10 ml of milliQ (prepare fresh)
- Proportions for 1 liter M-9 media
  - 200 ml of M-9 basis; 3 ml of $(NH_4)2SO_4$; 1 ml of $CaCl_2.2H_2O$; 1 ml trace elements; 20 ml glucose; 1ml $MgSO_4.7H_2O$; 1 ml $FeSO_4.7H_2O$ 2ml thiamine; 1ml antibiotic (standard conc.)

http://webzoom.freewebs.com/avikale/protocols%28culture%29/Recipe_for_M9_minimal_media.pdf

in silico M9 Minimal Media

This *in silico* media assumes the cell can uptake all the minerals wanted/needed from the media. It does not allow amino acid uptake.

Reaction Abbreviation	Reaction Name	Formula	Lower Bound	Upper Bound
EX_ca2(e)	Calcium exchange	ca2[e] <=>	-1000	1000
EX_cl(e)	Chloride exchange	cl[e] <=>	-1000	1000
EX_co2(e)	CO2 exchange	co2[e] <=>	-1000	1000
EX_cobalt2(e)	Co2+ exchange	cobalt2[e] <=>	-1000	1000
EX_cu2(e)	Cu2+ exchange	cu2[e] <=>	-1000	1000
EX_fe2(e)	Fe2+ exchange	fe2[e] <=>	-1000	1000
EX_fe3(e)	Fe3+ exchange	fe3[e] <=>	-1000	1000
EX_h(e)	H+ exchange	h[e] <=>	-1000	1000
EX_h2o(e)	H2O exchange	h2o[e] <=>	-1000	1000
EX_k(e)	K+ exchange	k[e] <=>	-1000	1000
EX_mg2(e)	Mg exchange	mg2[e] <=>	-1000	1000
EX_mn2(e)	Mn2+ exchange	mn2[e] <=>	-1000	1000
EX_mobd(e)	Molybdate exchange	mobd[e] <=>	-1000	1000
EX_na1(e)	Sodium exchange	na1[e] <=>	-1000	1000
EX_tungs(e)	tungstate exchange	tungs[e] <=>	-1000	1000
EX_zn2(e)	Zinc exchange	zn2[e] <=>	-1000	1000
EX_ni2(e)	Ni2+ exchange	ni2[e] <=>	-1000	1000
EX_sel(e)	Selenate exchange	sel[e] <=>	-1000	1000
EX_slnt(e)	selenite exchange	slnt[e] <=>	-1000	1000
EX_so4(e)	Sulfate exchange	so4[e] <=>	-1000	1000
EX_nh4(e)	Ammonia exchange	nh4[e] <=>	-1000	1000
EX_pi(e)	Phosphate exchange	pi[e] <=>	-1000	1000
EX_cbl1(e)	Cob(I)alamin exchange	cbl1[e] <=>	-0.01	1000

Monk, J. M., P. Charusanti, et al. (2013). "Genome-scale metabolic reconstructions of multiple Escherichia coli strains highlight strain-specific adaptations to nutritional environments." Proc Natl Acad Sci USA 110(50): 20338-20343.

Amino Acid Exchange Reactions

Rxn name	Rxn description	Formula	LB	UB
EX_ala_L(e)	L-Alanine exchange	ala-L[e] <=>	0	1000
EX_arg_L(e)	L-Arginine exchange	arg-L[e] <=>	0	1000
EX_asn_L(e)	L-Asparagine exchange	asn-L[e] <=>	0	1000
EX_asp_L(e)	L-Aspartate exchange	asp-L[e] <=>	0	1000
EX_cys_L(e)	L-Cysteine exchange	cys-L[e] <=>	0	1000
EX_gln_L(e)	L-Glutamine exchange	gln-L[e] <=>	0	1000
EX_glu_L(e)	L-Glutamate exchange	glu-L[e] <=>	0	1000
EX_gly(e)	Glycine exchange	gly[e] <=>	0	1000
EX_his_L(e)	L-Histidine exchange	his-L[e] <=>	0	1000
EX_ile_L(e)	L-Isoleucine exchange	ile-L[e] <=>	0	1000
EX_leu_L(e)	L-Leucine exchange	leu-L[e] <=>	0	1000
EX_lys_L(e)	L-Lysine exchange	lys-L[e] <=>	0	1000
EX_met_L(e)	L-Methionine exchange	met-L[e] <=>	0	1000
EX_phe_L(e)	L-Phenylalanine exchange	phe-L[e] <=>	0	1000
EX_pro_L(e)	L-Proline exchange	pro-L[e] <=>	0	1000
EX_ser_L(e)	L-Serine exchange	ser-L[e] <=>	0	1000
EX_thr_L(e)	L-Threonine exchange	thr-L[e] <=>	0	1000
EX_trp_L(e)	L-Tryptophan exchange	trp-L[e] <=>	0	1000
EX_tyr_L(e)	L-Tyrosine exchange	tyr-L[e] <=>	0	1000
EX_val_L(e)	L-Valine exchange	val-L[e] <=>	0	1000

Note: Amino acids are only allowed to be secreted in the basic model (LB = 0). The model can be modified to allow amino acid uptake.

No essential amino acids for *E.coli*

K-12 Undefined Media

- K-12 is an undefined media based on yeast extract and phosphates.
- Growth in K-12 media was simulated by adjusting lower bounds of exchange reactions to correspond to media conditions

	Chemical	Concentration
K12 Medium:	KH_2PO_4	2 g/L
	$K_2HPO_4.3H_2O$	4 g/L
	$(NH_4)_2HPO_4$	5 g/L
	Yeast Extract	5 g/L
	Glucose	25 g/L
	$MgSO_4.7H_2O$	0.5 g/L
	Thiamine	2.5 mg/L
	K12 trace metal	5 ml/L
K12 trace metal solution:	NaCl	5 g/L
	$ZnSO_4.7H_2O$	1 g/L
	$MnCl_2.4H_2O$	4 g/L
	$FeCl_3.6H_2O$	4.75 g/L
	$CuSO_4.5H_2O$	0.4 g/L
	H_3BO_3	0.575 g/L
	$NaMoO_4.2H_2O$	0.5 g/L
	6N H_2SO4	12.5 ml/L

Assigned Metabolite Uptake Rates for K-12 Media

(Metabolite lower bound determined by initial concentration of metabolite times the ratio of the initial glucose concentration/lower bound)

Metabolite	MW (g/mol)	g/L in media	mmol/L in media	Lower bound (mmol gDW^{-1}h^{-1})
Glucose	180.16	25	138.7655417	-11
Ammonium	18.03851	1.365829484	75.71742258	-6.002150375
Phosphate	94.9714	6.655736264	70.08147994	-5.555386948
Potassium	39.0983	1.945099309	49.74894838	-3.943619038
Sulfate	96.07	0.203323529	2.11641021	-0.167768684
Chloride	35.453	0.062050364	1.750214773	-0.138740225
Copper	63.546	0.000509009	0.008010093	-0.000634963
Iron (III)	55.845	0.00490684	0.087865335	-0.00696512
Magnesium	24.305	0.049304203	2.028562155	-0.160804933
Manganese	54.938044	0.005551956	0.101058487	-0.008010947
Molybdate	95.95	0.001826052	0.019031286	-0.001508618
Sodium	22.98976928	0.029775544	1.295164976	-0.102668246
Thiamine	265.35	0.0025	0.009421519	-0.000746848
Zinc	65.38	0.001136847	0.017388304	-0.001378378
Alanine	89.09	0.225	2.525535975	-0.200200247
Arginine	174.2	0.145	0.832376579	-0.065982824
Asparagine	132.12	0.16	1.211020285	-0.095998062
Aspartic acid	133.1	0.16	1.202103681	-0.09529124
Cysteine	121.16	0.02	0.165070981	-0.013085243
Glutamine	146.14	0.345	2.360749966	-0.187137594
Glutamic Acid	147.13	0.345	2.344865085	-0.185878393
Glycine	75.07	0.135	1.798321567	-0.14255367
Histidine	155.15	0.06	0.386722527	-0.030655649
Isoleucine	131.17	0.145	1.105435694	-0.08762833
Leucine	131.17	0.21	1.600975833	-0.126909995
Lysine	146.19	0.22	1.504890895	-0.119293303
Methionine	149.21	0.045	0.301588365	-0.02390703
Phenylalanine	165.19	0.12	0.726436225	-0.05758489
Proline	115.13	0.115	0.998870842	-0.079180891
Serine	105.09	0.13	1.237034922	-0.098060253
Threonine	119.12	0.135	1.133310947	-0.089838012
Tyrosine	181.19	0.09	0.496716154	-0.039374888
Valine	117.15	0.165	1.408450704	-0.111648451

For more accurate results these uptake rates need to be measured

Sarah Allred, "Metabolic Modeling of Spider Silk Production in *E. coli*," MS Thesis, USU, 2014