acetate_auxotrophy_model

April 4, 2023

1 Imports and setup

[ ]:
```
import escher  # flux maps with escher do not export well to html or pdf
import warnings
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from cameo import load_model
from cameo.flux_analysis.simulation import pfba
from cobra.manipulation.delete import find_gene_knockout_reactions
from cobra.io import save_json_model
from cobra.core import Metabolite, Reaction
from cobra.exceptions import Infeasible
```

Load model

[ ]:
```
# It appears the the iJO1366 model originally used for this work has been␣
 ↪updated to
# improve formatting and compatibility. Loading the old model format can be␣
 ↪problematic,
# so here the new one is used. Reaction names might not match perfectly with␣
 ↪maps since
# they were made for the original model.
iJO = load_model("iJO1366.json", sanitize=True)
iJO.solver = "glpk"
model = iJO.copy()
```

Tryptophanase is considered irreversible under physiological conditions

[ ]:
```
model.reactions.TRPAS2.lower_bound = 0
```

Check pfba solution fluxes on WT

[ ]:
```
pmap = escher.Builder(
    model=model,
    map_json="acetate_coupling_escherMap.json",
    reaction_data=pfba(model).fluxes,
```

```
    # use_3d_transformation=True,
    reaction_styles=["color", "size", "text", "abs"],
    reaction_scale=[
        {"type": "min", "color": "#c8c8c8", "size": 12},
        {"type": "mean", "color": "#9696ff", "size": 20},
        {"type": "max", "color": "#ff0000", "size": 25},
    ],
)
pmap
```

```
Builder(reaction_data={'EX_cm_e': 0.0, 'EX_cmp_e': 0.0, 'EX_co2_e': 19.
↪68710310273891, 'EX_cobalt2_e': -2.4552…
```

2 Acetate auxotrophy for consortia paper

2.1 Deletions and insertions:

2.1.1 Implement literature based deletions

Project deletions to the model as noted on the manuscript for acetate auxotrophy:

- aceEF > PDH
- focA > FORt2pp
- pflB > PFL
- poxB > POX

[ ]:
```
for rxn in [
    "PDH",
    "FORt2pp",
    "PFL",
    "POX",
]:
    model.reactions.get_by_id(rxn).knock_out()
```

Check if the strain grows without acetate

[ ]:
```
print("Growth rate:", round(model.optimize().objective_value, 2))

pmap = escher.Builder(
    model=model,
    map_json="acetate_coupling_escherMap.json",
    reaction_data=pfba(model).fluxes,
    # use_3d_transformation=True,
    reaction_styles=["color", "size", "text", "abs"],
    reaction_scale=[
        {"type": "min", "color": "#c8c8c8", "size": 12},
        {"type": "mean", "color": "#9696ff", "size": 20},
        {"type": "max", "color": "#ff0000", "size": 25},
    ],
```

```
)
pmap
```

```
Growth rate: 0.95
```

```
Builder(reaction_data={'EX_cm_e': 0.0, 'EX_cmp_e': 0.0, 'EX_co2_e': 21.
↪131228844027483, 'EX_cobalt2_e': -2.367…
```

Seems perfectly fine on simulation, but the the route to getting acetate is clearly different based on pFBA solution. In this case, acetate is sourced from PPP.

Following genes were selected to remove potential escapes in PFL, take out putative pyruvate oxidoreductase and eliminate a route to channel PPP carbon to acetyl-coa.

[ ]:
```
rxnList = find_gene_knockout_reactions(
    model,
    [
        model.genes.b3114,  # tdcE
        model.genes.b3951,  # pflD
        model.genes.b3952,  # pflC
        model.genes.b1378,  # pfo / previously ybdK
        model.genes.b4381,  # deoC
    ],
)
for rxn in rxnList:
    rxn.upper_bound = 0.0
    rxn.lower_bound = 0.0
```

Check if the strain grows without acetate

[ ]:
```
print("Growth rate:", round(model.optimize().objective_value, 2))

pmap = escher.Builder(
    model=model,
    map_json="acetate_coupling_escherMap.json",
    reaction_data=pfba(model).fluxes,
    # use_3d_transformation=True,
    reaction_styles=["color", "size", "text", "abs"],
    reaction_scale=[
        {"type": "min", "color": "#c8c8c8", "size": 12},
        {"type": "mean", "color": "#9696ff", "size": 20},
        {"type": "max", "color": "#ff0000", "size": 25},
    ],
)
pmap
```

```
Growth rate: 0.9
```

```
Builder(reaction_data={'EX_cm_e': 0.0, 'EX_cmp_e': 0.0, 'EX_co2_e': 23.
↪243431839008128, 'EX_cobalt2_e': -2.238…
```

Deletions generate an impact on growth rate of the mutant but doesn't eliminate it.

2 mechanisms, either related to serine or threonine can be used to generate acetate/acetyl-CoA.

Take out threonine route with 2 reaction deletions

[ ]:
```
rxnList = ["THRD", "THRA"]
for rxn in rxnList:
    model.reactions.get_by_id(rxn).knock_out()
```

[ ]:
```
print("Growth rate:", round(model.optimize().objective_value, 2))

pmap = escher.Builder(
    model=model,
    map_json="acetate_coupling_escherMap.json",
    reaction_data=pfba(model).fluxes,
    # use_3d_transformation=True,
    reaction_styles=["color", "size", "text", "abs"],
    reaction_scale=[
        {"type": "min", "color": "#c8c8c8", "size": 12},
        {"type": "mean", "color": "#9696ff", "size": 20},
        {"type": "max", "color": "#ff0000", "size": 25},
    ],
)
pmap
```

```
Growth rate: 0.85
```

```
Builder(reaction_data={'EX_cm_e': 0.0, 'EX_cmp_e': 0.0, 'EX_co2_e': 25.
↪17425507493298, 'EX_cobalt2_e': -2.1210…
```

Take out serine route with 1 reaction deletion

[ ]:
```
model.reactions.ETHAAL.knock_out()
```

[ ]:
```
print("Growth rate:", round(model.optimize().objective_value, 2))

pmap = escher.Builder(
    model=model,
    map_json="acetate_coupling_escherMap.json",
    reaction_data=pfba(model).fluxes,
    # use_3d_transformation=True,
    reaction_styles=["color", "size", "text", "abs"],
    reaction_scale=[
        {"type": "min", "color": "#c8c8c8", "size": 12},
        {"type": "mean", "color": "#9696ff", "size": 20},
        {"type": "max", "color": "#ff0000", "size": 25},
    ],
)
pmap
```

```
Growth rate: 0.0
```

```
Builder(reaction_data={'EX_cm_e': 0.0, 'EX_cmp_e': 0.0, 'EX_co2_e': 0.
↪7411764705882343, 'EX_cobalt2_e': 0.0, '…
```

At this point, model can not generate biomass anymore.

The last two were considered to be unlikely routes and not selected for deletion.

Confirm rescue with acetate.

[ ]:
```
model.reactions.EX_ac_e.bounds = (-2, 1000)

print("Growth rate:", round(model.optimize().objective_value, 2))

pmap = escher.Builder(
    model=model,
    map_json="acetate_coupling_escherMap.json",
    reaction_data=pfba(model).fluxes,
    # use_3d_transformation=True,
    reaction_styles=["color", "size", "text", "abs"],
    reaction_scale=[
        {"type": "min", "color": "#c8c8c8", "size": 12},
        {"type": "mean", "color": "#9696ff", "size": 20},
        {"type": "max", "color": "#ff0000", "size": 25},
    ],
)
pmap
```

```
Growth rate: 0.46
```

```
Builder(reaction_data={'EX_cm_e': 0.0, 'EX_cmp_e': 0.0, 'EX_co2_e': 7.
↪597114160428192, 'EX_cobalt2_e': -1.1492…
```

Adding acetate on this model enables biomass production.

3 Manipulations for xylose vs glucose catapolism

Project deletions to the model to shape sugar catabolism:

- xylA > XYLI1, XYLI2
- xylB > DXYLK, XYLK (with araB)
- araA > ARAI
- araB > XYLK (with xylB), XYLK2, RBK_L1
- ptsG > GLCptspp

[ ]:
```
for rxn in [
    "XYLI1",
    "XYLI2",
    "DXYLK",
    "ARAI",
```

```
    "XYLK",
    "XYLK2",
    "RBK_L1",
    "GLCptspp"
]:
    model.reactions.get_by_id(rxn).knock_out()
```

[ ]:
```
with model:
    model.reactions.EX_xyl__D_e.bounds = (-10, 1000)
    model.reactions.EX_glc__D_e.bounds = (0, 1000)
    model.reactions.EX_ac_e.bounds = (-2, 1000)
    print("Growth rate:", round(model.optimize().objective_value, 2))
    pmap = escher.Builder(
        model=model,
        map_json="acetate_coupling_escherMap.json",
        reaction_data=pfba(model).fluxes,
        # use_3d_transformation=True,
        reaction_styles=["color", "size", "text", "abs"],
        reaction_scale=[
            {"type": "min", "color": "#c8c8c8", "size": 12},
            {"type": "mean", "color": "#9696ff", "size": 20},
            {"type": "max", "color": "#ff0000", "size": 25},
        ],
    )
pmap
```

```
Growth rate: 0.04
```

```
Builder(reaction_data={'EX_cm_e': 0.0, 'EX_cmp_e': 0.0, 'EX_co2_e': 2.
↪4205664480498825, 'EX_cobalt2_e': -9.619…
```

Doesn't look like it. Just capable of using acetate alone for growth though.

4 Add isobutyrate production pathway

[ ]:
```
# kivD overexpression
IVDC = Reaction(
    id="IVDC",
    name="Alpha-ketoisovalerate decarboxylase",
    lower_bound=0,
    upper_bound=1000,
)
model.add_reaction(IVDC)
mppal_c = Metabolite(
    id="2mppal_c",
    name="2-methylpropanal",
    compartment="c",
    formula="C4H8O",
```

```
    charge=0,
)
model.add_metabolites(mppal_c)
model.reactions.IVDC.build_reaction_from_string("3mob_c + h_c --> 2mppal_c +␣
 ↪co2_c")
balance = model.reactions.IVDC.check_mass_balance()
if balance != {}:
    print(balance)

# padA overexpression
IBTADH = Reaction(
    id="IBTADH",
    name="isobutyraldehyde dehydrogenase",
    lower_bound=0,
    upper_bound=1000,
)
model.add_reaction(IBTADH)
ibt_c = Metabolite(
    id="ibt_c", name="isobutyrate", compartment="c", formula="C4H7O2", charge=-1
)
model.add_metabolites([ibt_c])
model.reactions.IBTADH.build_reaction_from_string(
    "2mppal_c + nad_c + h2o_c  --> ibt_c + nadh_c + 2 h_c"
)
balance = model.reactions.IBTADH.check_mass_balance()
if balance != {}:
    print(balance)

# ibt transport - assume free/non-beneficial
IBTT = Reaction(
    id="IBTT", name="isobutyrate transport", lower_bound=0, upper_bound=1000
)
model.add_reaction(IBTT)
ibt_e = Metabolite(
    id="ibt_e", name="isobutyrate", compartment="e", formula="C4H7O2", charge=-1
)
model.add_metabolites([ibt_e])
model.reactions.IBTT.build_reaction_from_string("ibt_c => ibt_e")
balance = model.reactions.IBTT.check_mass_balance()
if balance != {}:
    print(balance)

model.add_boundary(metabolite=model.metabolites.ibt_e, type="exchange", lb=0)
```

[ ]:
```
<Reaction EX_ibt_e at 0x14f20b610>
```

Check how the fluxes look with a little push towards isobutyrate production.

[ ]:
```
with model:
    model.reactions.EX_glc__D_e.bounds = (-4, 1000)
    model.reactions.EX_ac_e.bounds = (-2, 1000)
    model.reactions.EX_ibt_e.bounds = (0.2, 1000)
    print("Growth rate:", round(model.optimize().objective_value, 2))
    pmap = escher.Builder(
        model=model,
        map_json="acetate_coupling_escherMap.json",
        reaction_data=pfba(model).fluxes,
        # use_3d_transformation=True,
        reaction_styles=["color", "size", "text", "abs"],
        reaction_scale=[
            {"type": "min", "color": "#c8c8c8", "size": 12},
            {"type": "mean", "color": "#9696ff", "size": 20},
            {"type": "max", "color": "#ff0000", "size": 25},
        ],
    )
pmap
```

```
Growth rate: 0.4
```

```
Builder(reaction_data={'EX_cm_e': 0.0, 'EX_cmp_e': 0.0, 'EX_co2_e': 10.
↪624075296732311, 'EX_cobalt2_e': -1.009…
```

Eliminate acetate exchange for export.

[ ]:
```
model.reactions.EX_ac_e.bounds = (0, 1000)
```

5 Export model

[ ]:
```
save_json_model(model, filename="acetate_auxotroph.json")
```

6 Theoretical limits

Check maintenance free isobutyrate production limit

[ ]:
```
with model:
    model.reactions.ATPM.bounds = (0, 0)
    model.objective = model.reactions.EX_ibt_e.id
    model.reactions.EX_glc__D_e.bounds = (-5, 100)
    sol = pfba(model)

df = sol.fluxes.to_frame()
df[(df.index.str.startswith("EX_")) & (abs(df.fluxes) >= 0.001)]
```

[ ]:

	fluxes
EX_co2_e	10.0
EX_glc__D_e	-5.0

```
EX_h_e          5.0
EX_h2o_e       10.0
EX_o2_e        -5.0
EX_ibt_e        5.0
```