

Anexos – Relatório 2

Anexo 1

```
from reframed.io.sbml import load_cbmodel
from mewpy.simulation import get_simulator, set_default_solver

model = load_cbmodel('iNJ661_middlebrook_ficha2.xml', flavor='cobra')

simul = get_simulator(model)
print(simul.objective)
print(simul.summary())

result = simul.simulate()
print(result)

{'R_biomass_Mtb_9_60atp': 1.0}

Metabolites: 826
Reactions: 1022
Genes: 0
objective: 0.0521994690017773
Status: OPTIMAL
Constraints: OrderedDict()
Method: SimulationMethod.FBA
```

Anexo 2

```
# procurar metabolitos excretados em condições ambientais ótimas
import re
padrao = re.compile("EX_.*")
metabolitos = []
for k,v in result.fluxes.items():
    if padrao.search(k) != None and result.fluxes[k] > 0:
        metabolitos.append((k,v))
        print(k,v)

R_EX_glu_DASH_L_LPAREN_e_RPAREN_ 0.658793211920325
R_EX_h_LPAREN_e_RPAREN_ 2.6504300905765845
R_EX_h2_LPAREN_e_RPAREN_ 1.0745259648858694
R_EX_h2co3_LPAREN_e_RPAREN_ 0.9231070336098581
R_EX_h2o_LPAREN_e_RPAREN_ 2.102597342112813
R_EX_lac_DASH_L_LPAREN_e_RPAREN_ 0.7188478492329926
R_EX_succ_LPAREN_e_RPAREN_ 0.12643579561653812
```

Anexo 3

```
# potenciais reações alvo
essential_reactions = simul.essential_reactions()
print(len(essential_reactions))
```

```
# potenciais genes alvo
simul.essential_genes()

[]
```

Anexo 4

```
from mewpy.problems.reactions import RK0Problem

problem = RK0Problem(model, [])

# impacto do knockout de "R_AACPS11" no crescimento microbiano
c1 = problem.solution_to_constraints({"R_AACPS11":0})
result_ko1 = simul.simulate(method="FBA", constraints=c1)
print(result_ko1)

# impacto do knockout de "R_EX_glyc_LPAREN_e_RPAREN_" no crescimento microbiano
c2 = problem.solution_to_constraints({"R_EX_glyc_LPAREN_e_RPAREN_":0})
result_ko2 = simul.simulate(method="FBA", constraints=c2)
print(result_ko2)
# objective = 0 => não há crescimento microbiano

objective: -0.0
Status: OPTIMAL
Constraints: OrderedDict([('R_AACPS11', 0)])
Method:FBA

objective: -0.0
Status: OPTIMAL
Constraints: OrderedDict([('R_EX_glyc_LPAREN_e_RPAREN_', 0)])
Method:FBA
```

Anexo 5

```
# impacto do knockout de "R_AACPS11" nas restantes reações essenciais

dic_diff1 = {} # todos os valores de fluxo que foram alterados devido ao knockout
for r in essential_reactions:
    if result.fluxes[r] != result_ko1.fluxes[r]:
        dic_diff1[r] = [result.fluxes[r], result_ko1.fluxes[r]]

dic_menos1 = {} # fluxos que diminuíram para menos de um décimo do fluxo inicial
dic_mais1 = {} # fluxos que aumentaram pelo menos 10 vezes em relação ao fluxo inicial
num_menos1 = 0
num_mais1 = 0
for k,v in dic_diff1.items():
    if v[1] / v[0] < 0.1:
        dic_menos1[k] = f"{{(v[1] / v[0]) * 100:.2f}}%"
        num_menos1 += 1
    elif v[1] / v[0] > 10:
        dic_mais1[k] = f"{{(v[1] / v[0]) * 100:.2f}}%"
        num_mais1 += 1

print(f"fluxo_ko / fluxo_inicial < 0.1 -> {num_menos1}")
print(f"fluxo_ko / fluxo_inicial > 10 -> {num_mais1}")

fluxo_ko / fluxo_inicial < 0.1 -> 262
fluxo_ko / fluxo_inicial > 10 -> 23
```

Anexo 6

```
# impacto do knockout de "R_EX_glyc_LPAREN_e_RPAREN_" nas restantes reações essenciais

dic_diff2 = {} # todos os valores de fluxo que foram alterados devido ao knockout
for r in essential_reactions:
    if result.fluxes[r] != result_ko2.fluxes[r]:
        dic_diff2[r] = [result.fluxes[r], result_ko2.fluxes[r]]

dic_menos2 = {} # fluxos que diminuíram para menos de um décimo do fluxo inicial
dic_mais2 = {} # fluxos que aumentaram pelo menos 10 vezes em relação ao fluxo inicial
num_menos2 = 0
num_mais2 = 0
for k,v in dic_diff2.items():
    if v[1] / v[0] < 0.1:
        dic_menos2[k] = f"{{(v[1] / v[0]) * 100:.2f}}%"
        num_menos2 += 1
    elif v[1] / v[0] > 10:
        dic_mais2[k] = f"{{(v[1] / v[0]) * 100:.2f}}%"
        num_mais2 += 1

print(f"fluxo_ko / fluxo_inicial < 0.1 -> {num_menos2}")
print(f"fluxo_ko / fluxo_inicial > 10 -> {num_mais2}")

fluxo_ko / fluxo_inicial < 0.1 -> 269
fluxo_ko / fluxo_inicial > 10 -> 35
```