

Biotech Beer Brewing

Project status report

Dominik Schmidt
Jakob Wittmann

April 10, 2018

1 Project status

1.1 Milestones

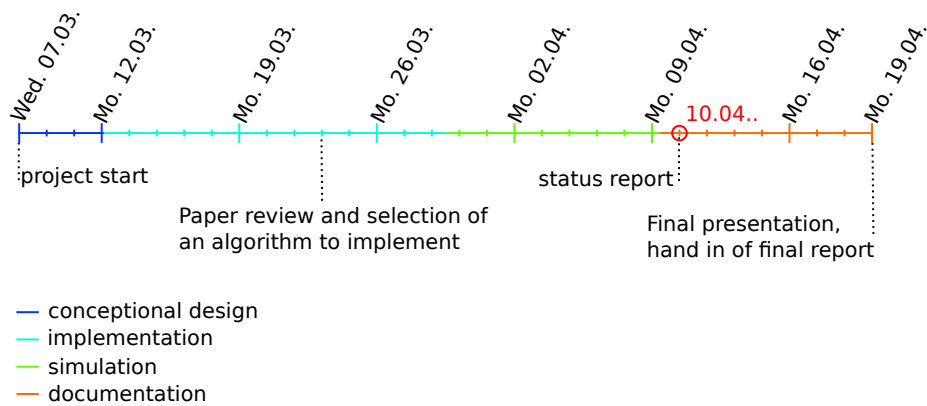


Figure 1: Project timeline

1.2 Progress

The progress of the main project processes in table 1 is illustrated in figure 2. The dashed colored lines show a simple estimated future progress of the processes. If the current trend holds on, the project will be finished approximately 26 working days after the final due date.

Table 1: Project processes

Process	Effort		start date	due date	progress
	Days	Percent			
conceptional design	3.1 d	10 %	07.03.	12.03.	100 %
implementation	12.4 d	40 %	12.03.	28.03.	92 %
research	4.65 d	15 %	12.03.	16.03.	100 %
concept	1.55 d	5 %	16.03.	20.03.	100 %
coding	6.2 d	20 %	20.03.	28.03.	75 %
simulation	7.75 d	25 %	28.03.	09.04.	42 %
research	4.65 d	15 %	28.03.	30.03.	75 %
setup	1.55 d	5 %	30.03.	02.04.	50 %
simulate setup	1.55 d	5 %	02.04.	09.04.	0 %
documentation	7.75 d	25 %	09.04.	17.04.	0 %
analyze results	1.55 d	5 %	09.04.	10.04.	0 %
prepare presentation	3.1 d	10 %	10.04.	16.04.	0 %
prepare report	3.1 d	10 %	16.04.	17.04.	0 %

2 Presentation of intermediate results

2.1 Simulation algorithm

This project will use parts of the Dynamic Multispecies Metabolic Modeling (DMMM) framework to implement dynamic flux balance analysis (DFBA). Table 2 compares selected features of the original implementation of DMMM with the implementation in this project.

Table 2: Overview of implemented features compared to DMMM

Feature	DMMM	This project
Model		
arbitrary many GEMs	yes	yes
arbitrary many metabolites in environment	yes	yes
mortality of bacteria	yes (in output flux)	yes
input/output flux of bacteria and metabolites	yes	no
parameterized initial state of environment composition	yes	yes
Michaelis-Menten kinetics	yes	yes
Algorithm		
ODE solver	yes	yes
different ODE solvers	yes	no
analytical solver	yes	no

As described by Zhuang et al. in [6] the algorithm uses a ODE solver with embedded FBA. A FBA is solved for each GEM in the model and for each time step in the discretised simulation time interval considering the changed metabolite and bacteria densities in the shared environment. The results of the

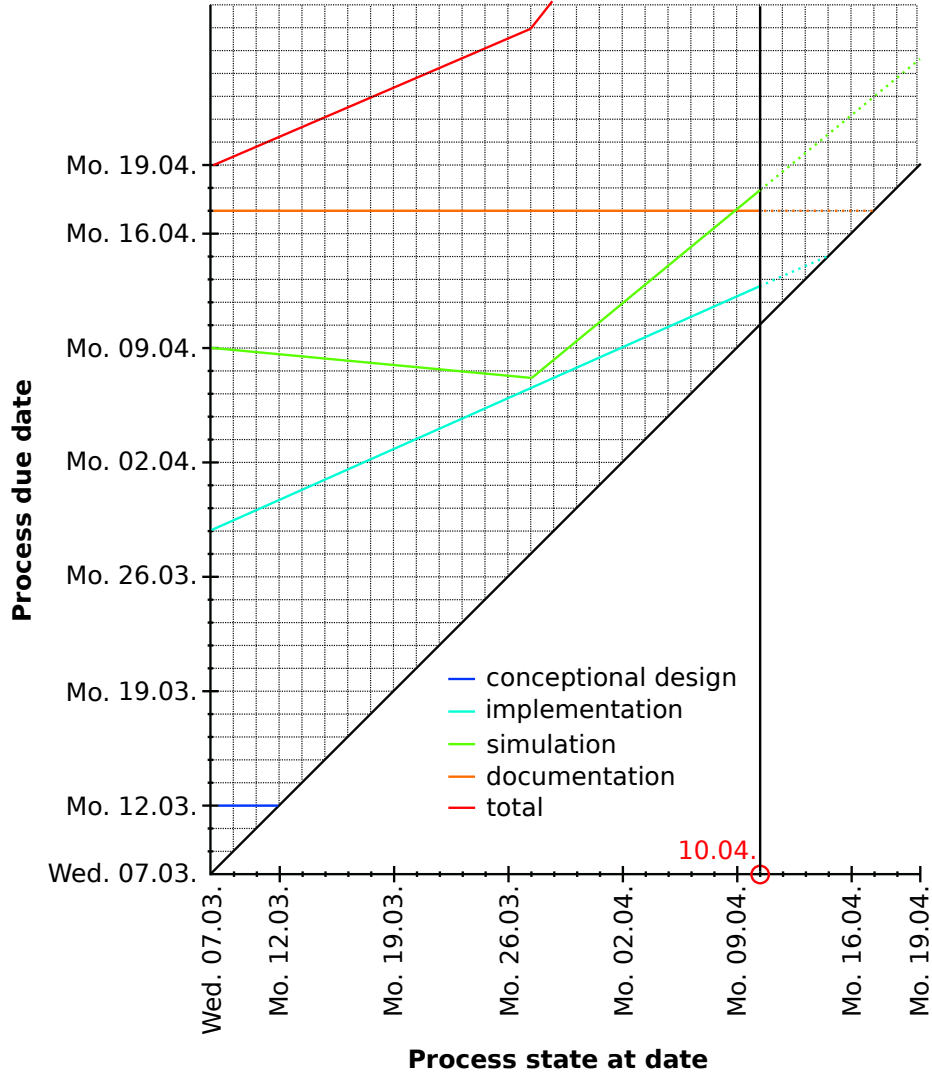


Figure 2: Project progress

FBA's are used by the ODE solver to solve the differential equations

$$\frac{dx_j}{dt} = \mu_j x_j \quad (1)$$

$$\frac{ds_i}{dt} = \sum_{j=1}^N v_{i,j} x_j \quad (2)$$

which models the dynamics of the bacteria's environment [5] where $i = 1 \dots N$ is the index of metabolites in the shared environment and $j = 1 \dots M$ is the index of bacteria. The bacteria density is modeled in x_j with $[x_j] = \frac{g}{l}$ and μ_j is the bacteria's growth rate with $[\mu_j] = \frac{mmol}{g_{DW}h}$. Input and output fluxes of the bacteria's models are modeled in $v_{i,j}$ with $[v_{i,j}] = \frac{mmol}{g_{DW}h}$, the densities of

metabolites in the shared environment in s_i with $[s_i] = \frac{mmol}{l}$.

In each time step each bacteria's metabolite intake must be changed dependent on the densities of the metabolites in the shared environment. To model saturation of metabolite intake for high metabolite densities Zhuang et al. implemented Michaelis-Menten kinetics [3]

$$v_{max,i,j} = \frac{v_{mm,i,j}s_i}{s_i + k_{mm,i,j}} \quad (3)$$

This formula describes the upper bound of the input flux $v_{max,i,j}$ for metabolite i of bacteria j dependent on the metabolite density s_i . The formula is characterized by two constants $[v_{mm,i,j}] = \frac{mmol}{g_{DWH}}$ and $[k_{mm,i,j}] = \frac{mmol}{l}$ for each bacteria and metabolite.

Mortality is considered using a constant $[\mu_{mort,j}] = \frac{mmol}{g_{DWH}}$ for each bacteria j in this implementation while Zhuang et al. modeled this using the output flux of bacteria out of the system.

Algorithm 1 shows a basic implementation of the differential equations solved by an ODE solver during the simulation similar to DMMM [6].

The algorithm expects a list of bacteria models consisting of

- GEM of this bacteria: A , v_{min} , v_{max} , w_{growth}
- v_{mm} (Michaelis-Menten V_{max}) for each exchange metabolite and species
- k_{mm} (Michaelis-Menten K) for each exchange metabolite and species
- mortality μ_{mort}

Furthermore a list of all exchange metabolites in the environment, the bacteria and metabolite densities.

In a first step the upper bounds of the intake fluxes are updated for each bacteria j and exchange metabolite i . The function `update_intake_bounds(modelj, sj, mi)` calculates the upper bounds using the formula 3 if the metabolite m_i is contained in `modelj` as a exchange metabolite and updates this value in the model.

In a next step the GEMs are optimized for growth using FBA, the results are used as growth rate μ_j and actual input and output fluxes v_j of bacteria j in this time step.

The mortality is considered by subtracting the constants μ from the growth rates μ .

At last step the slopes \dot{x} and \dot{s} are calculated according to 1 and 2 and returned to the ODE solver.

2.2 Simulation Setup

The goal of the simulation is to validate the basic functionality of the simulator using a simplified setup of a realistic future simulation scenario. As defined in our project goals, this simulation scenario is the dynamic flux balance analysis (DFBA) of a co-culture of *Saccharomyces cerevisiae* and *Lactobacillus plantarum*.

As genome-scale models a model of *Lactobacillus plantarum* published by Teusink et al. [4]. A decision about a yeast model is not made yet.

Algorithm 1: Differential equation with embedded FBA

```
1 function step(model1...modelM, m1...mN, x1...M, s1...sN);  
   Input : bacteria models modelj, exchange metabolites mi in  
           environment, bacteria densities xj, metabolite densities si  
   Output: slope of bacteria and metabolite densities  $\dot{x}_j, \dot{s}_i$   
2 for j := 1 to M do  
3   for i := 1 to M do  
4     | modelj := update_intake_bounds(modelj, sj, mi)  
5   end  
6 end  
7 for j := 1 to M do  
8   |  $\mu_j, v_j := FBA(model_j, w_{growth})$   
9 end  
10  $\mu := \mu - \mu_{mort}$   
11  $\dot{x} := diag(\mu) x$   
12 for j := 1 to M do  
13   for i := 1 to N do  
14     |  $\dot{s}[m_i] := \dot{s}[m_i] + v_j[m_i]x_j$   
15   end  
16 end  
17 return  $\dot{x}, \dot{s}$ 
```

Table 3: Model constants used in the simulation setup

Constant	S. cerevisiae	L. plantarum
Maximum glucose uptake rate (mmol/g/h)	18.5	?
Maximum oxygen uptake rate (mmol/g/h)	2.5	?
Glucose uptake saturation constant (g/l)	0.5	?
Oxygen uptake saturation constant (mM)	0.005	?

The simulation will consider two input metabolites: oxygen and glucose. Table 3 and table 4 contain all values needed to define the initial metabolite conditions and kinetics.

To verify the basic functionality of the simulator the resulting bacteria densities and metabolite densities of ethanol, d- and l-lactate, oxygen and glucose will be compared to existing data.

3 Outlook

The current delay of 26 days will be compensated with additional shifts and it is assumed that parts of both status reports can be reused in the final report.

The subsequent steps include:

- debugging of simulator code
- research on metabolite kinetics for *Lactobacillus plantarum*
- choosing a yeast GEM

Table 4: Simulation parameters used in the simulation setup

Parameter	value	reference
Initial glucose density (mmol/l)	153.031...690.697	TBD
Initial oxygen density (mmol/l)	0.5039	TBD [1]

- research on reference data to verify simulation results
- start simulations

References

- [1] Zdenek Bubnk et al. *Sugar technologists manual: chemical and physical data for sugar manufacturers and users*. Bartens, 1995.
- [2] *Chemical features of water*. 1987. URL: <http://www.fao.org/docrep/field/003/AC183E/AC183E04.htm>.
- [3] Kenneth A Johnson and Roger S Goody. “The original Michaelis constant: translation of the 1913 Michaelis–Menten paper”. In: *Biochemistry* 50.39 (2011), pp. 8264–8269.
- [4] Bas Teusink et al. “Analysis of Growth of *Lactobacillus plantarum* WCFS1 on a Complex Medium Using a Genome-scale Metabolic Model”. en. In: *Journal of Biological Chemistry* 281.52 (Dec. 2006), pp. 40041–40048. ISSN: 0021-9258, 1083-351X. DOI: 10.1074/jbc.M606263200. URL: <http://www.jbc.org/lookup/doi/10.1074/jbc.M606263200> (visited on 03/27/2018).
- [5] K. Zhuang et al. “The design of long-term effective uranium bioremediation strategy using a community metabolic model”. en. In: *Biotechnology and Bioengineering* 109.10 (Oct. 2012), pp. 2475–2483. ISSN: 00063592. DOI: 10.1002/bit.24528. URL: <http://doi.wiley.com/10.1002/bit.24528> (visited on 03/23/2018).
- [6] Kai Zhuang et al. “Genome-scale dynamic modeling of the competition between *Rhodospirillum rubrum* and *Geobacter* in anoxic subsurface environments”. en. In: *The ISME Journal* 5.2 (Feb. 2011), pp. 305–316. ISSN: 1751-7362, 1751-7370. DOI: 10.1038/ismej.2010.117. URL: <http://www.nature.com/articles/ismej2010117> (visited on 03/23/2018).

4 Appendix

$$d_{total} = 4.13 \frac{g}{l} \frac{g}{P} p + 997 \frac{g}{l} \quad (4)$$

$$d_{total} = \frac{m_{total}}{V_{total}} \quad (5)$$

$$d_{total} = \frac{m_{Glc} + m_W}{V_{total}} \quad (6)$$

$$d_{total} = \frac{m_{Glc} + d_W V_W}{V_{total}} \quad (7)$$

$$d_{total} = \frac{m_{Glc} + d_W (V_{total} - V_{Glc})}{V_{total}} \quad \text{suppose } V_{total} = V_{Glc} + V_W \quad (8)$$

$$d_{total} = \frac{m_{Glc} + d_W \left(V_{total} - \frac{m_{Glc}}{d_{Glc}} \right)}{V_{total}} \quad (9)$$

$$s_{Glc} = \frac{m_{glc}}{V_{total}} = \frac{d_{total} - d_W}{1 - \frac{d_W}{d_{Glc}}} \quad (10)$$

$$\begin{aligned} \text{oxygen saturation of water at } 20^\circ\text{C} &= 9.077 \frac{mg}{l} \quad [2] \quad s_{init} = 9.077 \frac{mg}{l} = \\ 9.077 \cdot 10^{-3} \frac{g}{l} &= \frac{9.077 \cdot 10^{-3} \frac{g}{l}}{18.015 \cdot 10^{-3} \frac{g}{mmol}} = 0,5039 \frac{mmol}{l} \end{aligned}$$