Biotech Beer Brewing Final report

Dominik Schmidt
Jakob Wittmann

***Abstract*—Bla blub**

***Index Terms*—key word, key word,...**

I. Introduction

Traditional beer brewing is done in a batch process by fermenting glucose to alcohol using Saccharomyces cerevisiae. In a first step wort is produced by mixing water, malt and hop and applying different enzymatical processes. As in a batch process the densities of metabolites in the culture are not controlled the fermentation product and so the quality of the beer is highly dependent on the composition of the wort. To enhance the product quality the process of fermentation must be understood in detail and correlations between the starting conditions and fermentation results must be found. This is typically done in experiments which are very time and cost intensive. Especially high effort is needed to reproduce starting conditions and if different yeast mutants or contamination by other bacteria shall be tested. These experiments are very costly and are not affordable for smaller breweries. A simulation approach to test different starting conditions will reduce the amount of experiments and so the costs and will enable development of new production methods also for smaller companies.

As the formulation of sufficient models for (1) and (2) depends on the production process, so the applied yeast and wort, this project will concentrate on the development of a simulation framework to enable the simulation of the fermentation products dependent on the fermentation's starting conditions.

II. Methods

Zomorri et al. summarizes in [12] models to predict the behavior of bacteria cultures and introduces different categories. Three of them are especially interesting to be used in this project: *steady-state models*, *spatio-temporal models* and *dynamic models*. *Steady-state models* like compartmentalized community-level metabolic modeling can not be used since a common objective can not be generally assumed, as it would be in a purely competitive co-cultures *Spatio-temporal models* have a very high computational effort as they take spacial and temporal varying bacteria densities into account. As the spacial aspect is not necessarily required in this project a more optimal approach shall be preferred. The remaining category of *dynamic models* is a well established method to simulate microbial co-cultures in batch processes and summarizes different extensions to dynamic flux balance analyses methods (DFBA) [12]. They use genome- scale models (GEM) to simulate the behavior of the bacteria cultures and add differential equations to model the external system dynamics.

Mehadevan et al. introduces two basic categories of DFBA approaches: *dynamic optimization approach* (DOA) and *static optimization approach* (SOA) [9]. In DOA a the linear programming problem (LP) which predicts the bacteria behavior is reformulized to a non-linear programming problem (NLP). This approach has a very high computational effort [5] compared to SOA and has only been used for relatively small GEMs with up to 13 modeled fluxes and 8 metabolites [8] [7].

Mehadevan et al. introduces SOA in [9] as follows: The simulation interval is divided into several intervals and the LP is solved for each of these time intervals dependent on the metabolite densities. The solution of the LP defines the bacteria growth and metabolite production at a certain point of time in the simulation time interval. These values are then used to solve the differential equations which models the external system dynamics. To solve the LP for the next time interval the new calculated, changed metabolite densities are used. This procedure is repeated until the end of the simulation time interval is reached. This approach makes use of the assumption that the cell internal dynamics are much faster than the external dynamics. In SOA the behavior of the bacteria is assumed to be constant during one time interval what leads to a linear approximation approach when solving the system of ordinary differential equations (ODE), similar to Euler-Cauchy methods.

Höffner et al. adds in [5] a further group, the *direct approach* (DA) which basically describes methods similar to SOA which uses an ODE solver instead of the Euler-Cauchy method. Due to the used ODE solver different numerical approximation methods can be used, not only the linear approximation. A good documented example for this group is the *Dynamic Multispecies Metabolic Modeling* framework by Zhuang et al. [10].

Henson et al. mentions a third group, *reformulation to a differential-glgebraic equation system* [4]. It shows also many similarities to SOA with the difference that the LP is reformulized but still solved as a LP embedded within the external ODE. The reformulated equation system makes it possible to enhance the efficiency of algorithm compared to SOA and DA [5].

The described DFBA methods in section **??** were rated based on the given information in the above mentioned papers, see table I.

DOA can not be used due to its high computational effort and medium-high implementation complexity. The approach

TABLE I: Rating of considered DFBA methods

Method	comp. effort	impl. complexity	flexibility
dynamic optimization approach (DOA)	high	medium-high	?
static optimization approach (SOA)	low	low	low
direct approach (DA)	medium	low	medium
reformulation to a differential-glgebraic equation system	low-medium	high	?

which uses *reformulation to a differential-glgebraic equation system* is currently available in matlab code and must be implemented in python in this project. Due to the high implementation complexity this approach will also be excluded. The remaining methods, SOA and DA, have similar ratings but as DA is more flexible as different ODE solvers can be used this approach seems more sustainable. Besides its flexibility the DA implementation DMMM by Zhuang et al. [10] can be publicly accessed and they provide a good documentation which will facilitate the implementation in this project.

TABLE II: 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	?

TABLE III: Simulation parameters used in the simulation setup

Parameter	value	reference
Initial glucose density (mmol/l)	272.9 ... 1230.755	equation 6, table V
Initial oxygen density (mmol/l)	0.5039	equation 7, table V

TABLE IV: Overview of implemented features compared to DMMM

Feature	DMMM	This project
Model		
arbitrary many GEMs	yes	yes
arbitrary many metabolites in environment	yes	yes
mortablility 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 [11] 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 FBAs are used by the ODE solver to solve the differential equations

$$\frac{\mathrm{d}x_j}{\mathrm{d}t} = \mu_j x_j \tag{1}$$

$$\frac{\mathrm{d}s_i}{\mathrm{d}t} = \sum_{j=1}^{N} v_{i,j} x_j \tag{2}$$

which models the dynamics of the bacteria's environment [10] where $i = 1...N$ is the index of metabolites in the shared environment and $j = 1...M$ 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 [6]

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

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 to constants $[v_{mm,i,j}] = \frac{mmol}{g_{DW}h}$ and $[k_{mm,i,j}] = \frac{mmol}{l}$ for each bacteria and metabolite.

Mortality is considered using a constant $[\mu_{mort,j}] = \frac{mmol}{g_{DW}h}$ 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 [11].

The algorithm expects a list of bacteria models consisting of

- GEM of this bacteria: A, $\boldsymbol{v_{min}}$, $\boldsymbol{v_{max}}$, $\boldsymbol{w_{growth}}$
- $\boldsymbol{v_{mm}}$ (Michaelis-Menten V_{max}) for each exchange metabolite and species
- $\boldsymbol{k_{mm}}$ (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 $update\_intake\_bounds(model_j, s_j, m_i)$ calculates the upper bounds using the formula 3 if the metabolite m_i is contained in $model_j$ 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 $\boldsymbol{v_j}$ of bacteria j in this time step.

The mortality is considered by subtracting the constants $\boldsymbol{\mu}$ from the growth rates $\boldsymbol{\mu}$.

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

III. Results

IV. Conclusions

Algorithm 1: Differential equation with embedded FBA

```
function step($(model_1...model_M, m_1...m_N, x_1...M, s_1...s_N)$);

Input  : bacteria models $model_j$, exchange metabolites
         $m_i$ in environment, bacteria densities $x_j$,
         metabolite densities $s_i$
Output: slope of bacteria and metabolite densities $\dot{x}_j, \dot{s}_i$
for $j := 1$ to $M$ do
    for $i := 1$ to $M$ do
        $model_j :=$
          $update\_intake\_bounds(model_j, s_j, m_i)$
    end
end
for $j := 1$ to $M$ do
    $\mu_j, \boldsymbol{v_j} := FBA(model_j, \boldsymbol{w_{growth}})$
end
$\boldsymbol{\mu} := \boldsymbol{\mu} - \boldsymbol{\mu_{mort}}$
$\dot{\boldsymbol{x}} := diag(\boldsymbol{\mu})\,\boldsymbol{x}$
for $j := 1$ to $M$ do
    for $i := 1$ to $N$ do
        $\dot{\boldsymbol{s}}[m_i] := \dot{\boldsymbol{s}}[m_i] + \boldsymbol{v_j}[m_i]x_j$
    end
end
return $\dot{\boldsymbol{x}}$, $\dot{\boldsymbol{s}}$
```

APPENDIX A

The following approximation is used to convert °C (“degree plato”) to a density measure (g/l) [3].

$$d_{total} = 4.13 \frac{g}{l\ {}^{\circ}P} p + 997 \frac{g}{l} \tag{4}$$

As the simulation framework expects metabolite densities relative to the total volume of the solution (mmol of metabolite per liter solution, mmol/l) the total density d_{total} must to converted to a density s_{glc}. It is assumed that $V_{total} = V_{glc} + V_W$.

$$d_{total} = \frac{m_{total}}{V_{total}}$$

$$d_{total} = \frac{m_{glc} + m_w}{V_{total}}$$

$$d_{total} = \frac{m_{glc} + d_w V_w}{V_{total}}$$

$$d_{total} = \frac{m_{glc} + d_w \left(V_{total} - V_{glc}\right)}{V_{total}}$$

$$d_{total} = \frac{m_{glc} + d_w \left(V_{total} - \frac{m_{glc}}{d_{glc}}\right)}{V_{total}}$$

$$s_{glc} = \frac{m_{glc}}{V_{total}} = \frac{d_{total} - d_w}{1 - \frac{d_w}{d_{glc}}} \tag{5}$$

Combining equation 4 and 5, including all constants and converting it to mmol/l leads to:

$$s_{glc} = \left(63.857 \frac{1}{{}^{\circ}P} p - 46.385\right) \frac{mmol}{l} \tag{6}$$

TABLE V: Constants used in this document

Constant	symbol	value	reference
Oxygen saturation of water at 20°C (mg/l)	-	9.077	[1]
Molar mass of water (g/mol)	-	18.015	[2]
Molar mass of glucose (g/mol)	-	180.156	[2]
Density of water (g/l)	d_w	1.00	[2]
Density of glucose (g/l)	d_{glc}	1.56	[2]
Typical glucose/water solution density to brew beer (°P)	-	5...20	-

To calculate the initial oxygen density in the solution it is assumed that the solution is at 20 °C and fully saturated with oxygen:

$$s_{init,ox} = 9.077 \frac{mg}{l} = 9.077\ 10^{-3} \frac{g}{l} = \frac{9.077\ 10^{-3} \frac{g}{l}}{18.015\ 10^{-3} \frac{g}{mmol}} = 0,5039 \frac{mmol}{l} \tag{7}$$

APPENDIX B

Appendix two text goes here.

REFERENCES

[1] “Chemical features of water,” 1987. [Online]. Available: http://www.fao.org/docrep/field/003/AC183E/AC183E04.htm

[2] “Pubchem: Open chemistry database,” 2018. [Online]. Available: https://pubchem.ncbi.nlm.nih.gov/

[3] Z. Bubnk, P. Kadlec, D. Urban, and M. Bruhns, *Sugar technologists manual: chemical and physical data for sugar manufacturers and users*. Bartens, 1995.

[4] M. A. Henson and T. J. Hanly, “Dynamic flux balance analysis for synthetic microbial communities,” *IET Systems Biology*, vol. 8, pp. 214–229, 2014.

[5] K. Höffner, S. M. Harwood, and P. I. Barton, “A reliable simulator for dynamic flux balance analysis,” *Biotechnology and Bioengineering*, vol. 110, no. 3, pp. 792–802, 2013.

[6] K. A. Johnson and R. S. Goody, “The original michaelis constant: translation of the 1913 michaelis–menten paper,” *Biochemistry*, vol. 50, no. 39, pp. 8264–8269, 2011.

[7] R. Luo, H. Wei, L. Ye, K. Wang, F. Chen, L. Luo, L. Liu, Y. Li, M. J. C. Crabbe, L. Jin, Y. Li, and Y. Zhong, “Photosynthetic metabolism of c3 plants shows highly cooperative regulation under changing environments: A systems biological analysis,” *Proceedings of the National Academy of Sciences*, vol. 106, pp. 847–852, 2009.

[8] R.-Y. Luo, S. Liao, G.-Y. Tao, Y.-Y. Li, S. Zeng, Y.-X. Li, and Q. Luo, “Dynamic analysis of optimality in myocardial energy metabolism under normal and ischemic conditions,” *Molecular Systems Biology*, vol. 2, 2006.

[9] R. Mahadevan, J. S. Edwards, and F. J. Doyle, “Dynamic Flux Balance Analysis of Diauxic Growth in Escherichia coli,” *Biophysical Journal*, vol. 83, pp. 1331–1340, 2002.

[10] K. Zhuang, E. Ma, D. R. Lovley, and R. Mahadevan, “The design of long-term effective uranium bioremediation strategy using a community metabolic model,” *Biotechnology and Bioengineering*, vol. 109, pp. 2475–2483, 2012.

[11] K. Zhuang, M. Izallalen, P. Mouser, H. Richter, C. Risso, R. Mahadevan, and D. R. Lovley, “Genome-scale dynamic modeling of the competition between rhodoferax and geobacter in anoxic subsurface environments,” *The ISME Journal*, vol. 5, pp. 305–316, 2011.

[12] A. R. Zomorrodi and D. Segrè, “Synthetic ecology of microbes: Mathematical models and applications,” *Journal of Molecular Biology*, vol. 428, pp. 837–861, 2016.