Effective Dynamic Models of Metabolic Networks

Michael Vilkhovoy*, Mason Minot†, Jeffrey D. Varner*
*School of Chemical Engineering, Purdue University, West Lafayette, IN 47907 USA †School of Chemical and Biomolecular Engineering, Cornell University, Ithaca, NY 14850 USA

***Abstract*—Mathematical models of biochemical networks are useful tools to understand and ultimately predict how cells utilize nutrients to produce valuable products. Modified version of the last paragraph of the introduction. Mike made this change.**

***Index Terms*—Dynamic metabolic models, flux balance analysis, cybernetic models**

I. Introduction

Three paragraphs, all must fit (along with abstract) on the first page. First, introduce the need to model and unstructured, structured, cybernetic, and recent palsson dynamic models of biochemical networks. Second paragraph introduce constraints based modeling, and its modifications along with EMs. Stress the issues with calculating EMs for large networks. Ref the parallel work. Third paragraph summarize what you have done. This paragraph must start with "In this study, we developed a ... ". It describes what you did, what the results were and what big picture conclusions you have drawn from the study. Last like should start with "Taken together, ... ". Always past tense. We did this, we saw that. SIMPLE LANGUAGE, SHORT DECLARATIVE SENTENCES.

The advent of new and creative approaches to the mathematical modeling of cell metabolism allows for the accurate description and prediction of cell behavior for a wide variety of organism and culture conditions. Monod kinetics have traditionally been used as unstructured models to describe cell growth, substrate consumption, and product formation [REF]. Structured models of metabolic pathways, such as Flux Balance Analysis (FBA), attempt to describe cellular behavior using the stoichiometry of biochemical networks, a pseudo-steady state assumption, an objective function, and system constraints [REF]. Cybernetic modeling approaches cell metabolism as a resource optimization problem; control variables are used to direct the allocation of resources through the cell in a manner most favorable to cell survival, growth, and product formation (Ramkrishna & Song, 2012). Dynamic Flux Balance Analysis couples FBA with dynamic or static optimization methods to describe the change in metabolic fluxes over time (Mahadevan et al. 2002).

II. Results

In the general case model, we demonstrated that the performance of the hybrid cybernetic model with FBA modes was equivalent to that of the elementary modes. In the proof of concept metabolic network consisting of 6 metabolites and 7 reactions (Fig. 1 A), METATOOL generated 6 elementary modes. FBA generated 3 possible modes (pathways) for the same metabolic network. Temporal profiles of the extracellular metabolites and artificial biomass concentration are presented for both the HCM FBA and HCM EM models (Fig. 1 B). HCM FBA has 9 kinetic parameters whereas HCM EM has 15 kinetic parameters. The kinetic parameters of HCM FBA were varied to fit the concentration profiles of the HCM EM model.

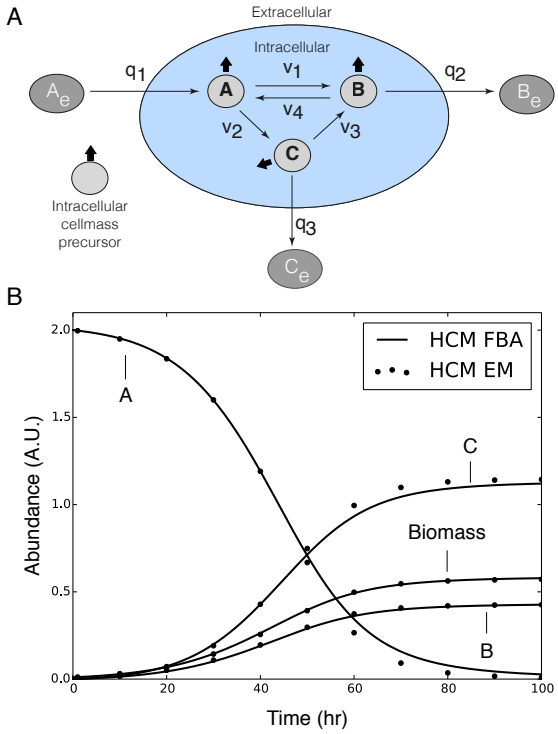

Fig. 1. A: Proof of concept metabolic network with six metabolites and seven reactions. Intracellular cellmass precursors A, B and C are balanced (no accumulation) while the extracellular metabolites (A_e, B_e and C_e) are not balanced (can accumulate). The blue-oval denotes the cell boundary, where q_j denotes the jth flux across the boundaries ([mmol/gdw-hr]) and v_k denotes the kth intracellular flux. B: Simulation of the extracellular metabolites using the FBA hybrid cybernetic approach (solid line) versus elementary modes (points) for the proof of concept metabolic network. FBA modes have comparable model performance compared to elementary modes.

General case = we showed that EMs and FBA models were equivalent for the small hypothetical model

Small E. coli case = we should that EMs and a fewer number of FBA models gave similar model performance

Large E. coli case = too many EMs, small number of FBA modes allowed us to simulate this network opening up the possibility of genome scale cybernetic models.

Sensitivity results = simple systematic method to eliminate

Manuscript received December 1, 2012; revised September 17, 2014. Corresponding author: J. Varner (email:jdvarner@purdue.edu).

FBA modes to give the smallest number needed, without complex weighting schemes.

In all cases, the first line of the paragraph states the result described in the paragraph. All results are in past tense.

III. Discussion

Three paragraphs. First paragraph, modified version of the last paragraph of the introduction, more details.

Second paragraph, contrast what we have done in the context of the field, in particular compare the hybrid cybernetic models, constraints based models and dynamic models of Palsson. Why is this study innovative?

Last paragraph, what is wrong with this study and where do we go from here? For example, could we extend the work done here to animal cells producing biologics, could we extend this to genome scale.

IV. Materials and Methods

Description of the method: Describe the method mathematics, use index form if possible, I the vector nomenclature to be hard to follow. Define all terms and symbols.

Estimation of model parameters Modify this section: Model parameters were estimated by minimizing the difference between simulations and experimental thrombin measurements (squared residual):

$$\min_{\mathbf{k}} \sum_{\tau=1}^{\mathcal{T}} \sum_{j=1}^{\mathcal{S}} \left(\frac{\hat{x}_j(\tau) - x_j(\tau, \mathbf{k})}{\omega_j(\tau)} \right)^2 \tag{1}$$

where $\hat{x}_j(\tau)$ denotes the measured value of species j at time τ, $x_j(\tau, \mathbf{k})$ denotes the simulated value for species j at time τ, and $\omega_j(\tau)$ denotes the experimental measurement variance for species j at time τ. The outer summation is with respect to time, while the inner summation is with respect to state. We minimized the model residual using Particle swarm optimization (PSO) [1]. PSO uses a *swarming* metaheuristic to explore parameter spaces. For each iteration, particles in the swarm compute their local error by evaluating the model equations using their specific parameter vector realization. From each of these local points, a globally best error is identified. Both the local and global error are then used to update the parameter estimates of each particle using the rules:

$$\mathbf{\Delta}_i = \theta_1 \mathbf{\Delta}_i + \theta_2 \mathbf{r}_1 (\mathcal{L}_i - \mathbf{k}_i) + \theta_3 \mathbf{r}_2 (\mathcal{G} - \mathbf{k}_i) \tag{2}$$

$$\mathbf{k}_i = \mathbf{k}_i + \mathbf{\Delta}_i \tag{3}$$

where $(\theta_1, \theta_2, \theta_3)$ are adjustable parameters, $\mathcal{L}_i$ denotes the local best solution found by particle i, and $\mathcal{G}$ denotes the best solution found over the entire population of particles. The quantities r_1 and r_2 denote uniform random vectors with the same dimension as the number of unknown model parameters ($\mathcal{K} \times 1$). In thus study, we used $(\theta_1, \theta_2, \theta_3) = (1.0, 0.05564, 0.02886)$. The quality of parameter estimates was measured using goodness of fit (model residual). The particle swarm optimization routine was implemented in the Python programming language.

References

1. J Kennedy and R Eberhart. Particle swarm optimization. In *Proceedings of the International Conference on Neural Networks*, pages 1942 – 1948, 1995.