# Control Challenges in Synthetic Biology \*

Christopher V. Rao<sup>\*</sup>

\* Department of Chemical and Biomolecular Engineering, Institute for Genomic Biology, University of Illinois at Urbana-Champaign, Urbana, IL 61810 USA (e-mail: cvrao@illinois.edu).

**Abstract:** Automation is increasingly being employed in the life sciences. New control problems are arising as a result, few with simple off-the-shelf solutions. This paper discusses some of the scheduling and control problems associated with automation in synthetic biology. It specifically focuses on the challenges associated with robotics, drawing heavily from our own experiences at the University of Illinois at Urbana-Champaign. No solution are presented and only the problems discussed. The goal is to motivate research in the process systems engineering community to solve problems in this new field.

*Keywords:* Synthetic Biology, Genetic Engineering, Control, Scheduling, Supply Chain, Robotics

# 1. INTRODUCTION

Our ability to engineer organisms has advanced dramatically over the past ten years. We now possess the ability to rewrite an organism's chromosome, either through modifying existing sequences or introducing entirely new ones. From these advances is emerging a new engineering discipline, commonly referred to as synthesis biology, focused on the *purposeful* redesign of exiting biological systems and ultimately the *de novo* design of entirely new ones (Way et al., 2014). While genetic engineering is not new, our capability to do it at large scale is. In fact, it is not unreasonable to envision a not-to-distant future where one programs organisms no differently than computers (Clancy and Voigt, 2010). To a limited degree, this is already happening in some labs (Gibson et al., 2010). Obviously, the potential impact is huge when one considers applications, with notable examples already in food, health, and energy (Xue et al., 2013; Ro et al., 2006; Atsumi et al., 2008; Schirmer et al., 2010). So are the potential ethical and safety problems that these new technologies pose.

Control plays a central role in synthetic biology (just as it does in most engineering disciplines). All living processes are tightly regulated using a variety of different feedback control mechanisms. Reengineering an organism, say to produce a chemical that it does not normally make, often requires reconfiguring its native control systems. These designs represent disturbances to an organism, and the robustness of its native control systems often determines success or failure. Clearly, one needs to adopt a systems perspective and consider regulation in any design (Lee et al., 2012). This is one reason why most designs have been implement in model organisms of low complexity such as *Escherichia coli* and *Saccharomyces cerevisiae*  (e.g. Baker's yeast). Even then our knowledge of theses organisms' physiology is still limited, and successful designs often require numerous iterative cycles coupled with classic genetic techniques such as selection and breeding. Significant effort, therefore, has focused on unraveling these native control systems so that we can ultimately reprogram them. Numerous example exist where control theory has played an instrumental role in our understanding of these native systems (Rao and Arkin, 2001).

Control is also starting to become an important tool in the synthetic biologist's toolbox (Chen et al., 2013). All organisms perform computations. Even the simplest organisms need to sense and respond to changes in their own state and external environment. Many examples now exist where entirely new logic has been introduced into cells using genetic analogues of logic gates (e.g. Moon et al. (2012)). The next step will be to introduce feedback control systems. In particular, when we introduce new processes in into organisms, such as a metabolic pathway to produce a chemical, we will also include control systems that regulate their production. Applying concepts from control engineering to synthetic biology is still an emerging area of research. One open challenge is that any controller ultimately needs to be instantiated using biochemical reactions. Not only does this limit what can be done but also introduces many new variables to the design equation (e.g. Wu and Rao (2010)).

The examples above briefly highlight some potential opportunities of control in synthetic biology. They address control in the purely biological sense, either by identifying or introducing feedback control loops at the genetic/molecular level. Many excellent reviews have been published in these areas, and the reader is directed to them for future detail (space limitations unfortunately prevent a detailed listing here). The focus of this paper, however, is to highlight opportunities in synthetic biology in the nonbiological sense. Specifically, I wish to highlight some of the control challenges we currently face at the University

<sup>\*</sup> This work is supported in part by grants from the National Institutes of Health, the Department of Energy, the Energy Biosciences Institute, the Defense Advanced Research Projects Agency, and the Roy J. Carver Charitable Trust.

of Illinois at Urbana-Champaign (UIUC). While some of these challenges are unique to UIUC, others are quite general and reflect the evolution of the field. I will specifically focus on the challenges associated with robotics, drawing heavily on our own experiences from a system developed at UIUC. Here, the issue is not biology *per se* but rather the lack of off-the-shelf solutions for these control problems. No solutions are presented – rather, my goal is to highlight a few significant control problems arising in automated synthetic biology.

# 2. ENABLING TECHNOLOGIES FOR SYNTHETIC BIOLOGY

Despite the impressive advances, genetic engineering is still challenging, particularly when done at large scale. Yes, we can write DNA – but synthesizing large sequences of DNA is expensive, inefficient, and time consuming. Even then, we are limited in our ability to rationally redesign biological systems. For one, we do not fully understand the syntax or semantics of this new programming language. In addition, we are almost always redesigning an existing system, for which our knowledge is often limited. As a consequence, design involves much trial and error.

A common approach is to employ combinatory design strategies. For example, if one does not know the DNA sequence necessary to activate the transcription of a gene, then a common strategy is to try all possible combinations. To the outsider, these approaches may look naïve – if we have millions of monkeys banging on typewriters, then surely one will produce something legible. They also admittedly highlight deep gaps in our understanding of basic biological processes. However, the power of biology is that one can often find the needle in the haystack of these large combinatorial libraries through clever experimental design and high-through measurement technologies such as flow cytometry or imaging mass spectrometry. The real challenge is the labor involved in assembling large combinatorial libraries of different designs, particularly when multiple factors need to be varied in a systematic manner.

Combinatorial design is where robotics becomes most practical and in many cases essential. Building these large libraries is tedious and laborious. It involves many repetitive tasks principally associated with precisely pipetting small values of liquid ( $0.5 - 50 \ \mu$ L). Many laboratories now employ liquid-handling robots to perform these tasks. The reactions are easy to multiplex using microtiter plates, containing 96 or more wells. The matrix-like structure of these plates makes it easy for robotic systems to precisely transfer liquids among different wells, often in groups using multichannel pipettes.

Liquid handling is still only one step in the process. You also need to amplify or synthesize specific sequences of DNA, assemble them into larger constructs, validate the assembly, introduce the DNA constructs into cells, and test the results. These other steps are rarely automated, even though they are equally tedious and laborious.

We have recently constructed in collaboration with Thermo Fisher Scientific an integrated robotic system at UIUC for automated DNA assembly and transformation. The



Fig. 1. Picture of robotic system at UIUC. The platform (horizontal) dimensions are  $6 \times 2$  m. The robotic arm in the pictured configuration is approximately 0.7 m tall.

goal of this system is to automate the entire synthetic biology design process in simple organisms such as bacteria or yeast. The system was originally conceived by Huimin Zhao, a professor at UIUC, and the author in 2012 and subsequently built in 2014. During the period, many individuals (too many to name) with diverse expertise in biology, robotics, industrial engineering, and analytic chemistry have contributed to the system.

The system is optimized for performing reactions in 96well microtiter plates. As shown in Figure 1, the systems features a 6 degree-of-freedom robotic arms that travels on a 5 meter long track and transfers microtiter plates among more than twenty instruments on the platform. These instruments are all computer-controlled using a common software platform. In some cases, the controllers for these instruments are quite complex and capable for executing complex series of tasks. The instruments include:

- **Liquid handlers** To add liquid to individual wells on microtiter plates or transfer liquid between individual wells on the same or different plates.
- **Thermocyclers** To perform the polyermase chain reaction (PCR) or any other reaction requiring precise temperature control (e.g. Gibson assembly (Gibson et al. (2009))).
- Shaking incubators To grow cells in liquid media.
- Fluorescent plate reader To measure DNA and cell concentrations.
- Magnetic bead separator To isolate DNA from cells.
- **Barcode labeller and reader** To label and track individual plates.
- Automated carousel To load, store, and remove plates from the system.

In addition, there are centrifuges, refrigerators, shaking and heating blocks, and plate sealers and peelers. Basically, the systems includes all of the operations necessary to synthesize large sequences of DNA and then transform them into cells. The system can, in theory, run continuously and autonomously for long periods of time (~weeks) without human intervention save for replenishing supplies. In addition, the system can run multiple jobs concurrently



Fig. 2. Diagram of scheduling problem. The colored boxes are used to represent the order of operations for a given job.

and in an asynchronous manner. The system is among the first of its kinds and has already been employed for a number of different applications. Potentially, it can be used to building thousands of design per day. While in some specialized applications this throughput can be achieved, in most others it still cannot for reasons elaborated below. An early demonstration video is available at https:// www.youtube.com/watch?v=Hwb735qZ-IQ.

While the robotic system has all of the ingredients necessary for automated design, it has presented us with a number of challenges associated with control, few that we fully appreciated when first designing the system. Before explaining these challenges, I will first state what has not been a challenge. One is controlling the robotic arm. It is relatively simple to train and program the arm, as the principal task involves moving plates of defined dimensions from one position to another. The other is speed, again with regards to the arm. Assembling DNA takes times, with many reactions taking hours to perform. Typically, each design takes at least one day to build with many steps taking hours each. Whether the robot takes 10 seconds or a few minutes to move a plate from one instrument to the next is irrelevant from our perspective.

Where we have struggled in realizing the full potential of our system is when dealing with issues related to scheduling control (Figure 2) and supply-chain management. The key challenges are listed below

- (1) Scheduling concurrent tasks
- (2) Error correction and control
- (3) Supply-chain control and data management
- (4) Resource allocation

and further elaborated in the next sections.

#### 3. SCHEDULING

The principal challenge with scheduling is that each job is unique, requiring a different set and sequence of operations to complete. Every design requires a unique program. While the general workflows are similar, the specifics vary from design to design. For example, we routinely employ just a few protocols for assembling DNA and then transforming it into cells. Once the reactions are setup, it is simply a matter of executing a specific protocol with a defined sequence of steps each utilizing a single instrument, such as a thermocycler for a PCR step or a heating block



Fig. 3. Example of 96-well microtiter plate. The dimensions of the plate are  $127.8 \times 85.5$  mm. The volume of each well is 350  $\mu$ L with a working volume of 100  $\mu$ L.

for a ligation step. These protocols are easy to program as one need only specify a predefined sequence of operations. The scheduling problem would be easy except for the numerous challenges discussed below.

The most immediate challenge is setting up these reactions. Here we need to combine different material into a single well on a microplate using the liquid-handling systems. As each design is unique, the material added to different wells is often unique as well. Moreover, for larger designs, it is not uncommon to combine material from over ten different sources, potentially originating from over ten different plates. These plates need to be delivered to the liquid handling system, which has finite storage capacity (approximately twenty plates). In addition, there are constraints on the order in which material can be added to individual wells. These constraints may reflect limitations of the liquid handling system: it cannot accurately add small volumes (< 1  $\mu$ L) to empty wells. Or, they may represent reaction constraints. For example, you always add the enzymes last. Once they are added, the plate needs to be maintained at 4°C by placing on a cooling block until one is ready to start the reaction, typically by heating to higher temperature by transferring the plate to a heating block or thermocycler.

While liquid-handling robots are now routinely used in laboratories, they are typically run as batch jobs on similar problems with numerous manual interventions. In addition, they are run with predefined starting conditions. For example, the different materials to be combined will always be placed in the same well positions on a plate loaded at exactly the same deck position. Also, the materials will always be combined at the same well position on a different plate (or position on it), again loaded always at the same deck position. This is impractical for a fully autonomous system as it reduces flexibility when running multiples jobs concurrently. Also, fixing the starting conditions for plates presents an equally challenging scheduling problem when each design potentially requires unique set of liquidating handling operations.

A second challenge reflects the use of containers, specifically the microtiter plates (Figure 3). Each design is contained within a single well, though not necessarily the same one, through the assembly process. When we transfer a plate between instruments, we transfer everything contained within that plate, potentially representing dozens of designs. This means that many operations run on the plates, such as heating or cooling, affect all wells on it. As the (bio)chemical reactions used to modify DNA need to be precisely controlled, we need to ensure that key steps in the assembly process are synchronized across the entire plate. Furthermore, many operations are potential bottlenecks in the process due to limited capacity. As an example, we do not wish to run the one of the thermocyclers on a plate containing just one design as this could tie up an instrument for many hours. Clearly, it would be preferable to run it on a full plate. However, this means that we need to co-locate designs requiring identical sequences of operations.

The third challenge involves the plug-and-play architecture of the robotic system, which may be unique to our design. We intentionally use off-the-shelf instruments from major vendors for which device drivers exist. This makes it easy to add new instruments or remove old one without having to change the underlying control software or worry about the intricacies of instrument operation. However, this architecture limits our ability to directly control these instruments in real time. In our current system, we can tell an instrument to run a specific program. The program can be complex, including conditionals and feedback loops. However, the instrument is locked into running that program until complete from the perspective of the overall system. In other words, there is no *real-time* bidirectional communication between the overall system and individual instruments. While it is possible to artificially introduce this communication by employing various programming hacks, these strategies are difficult to implement, do not scale well, and prone to error. This means, for example, that when scheduling complex operations on the liquid handler, typically where the problems are most complex, the operational sequence needs to be formulated as an individual (or lumped) step in relation to the overall control systems. It also means that we cannot add or remove plates from the liquid handler when it is a running a program. The arm needs to deliver the plates, have the liquid handler do its thing, and only then have the arm remove the plates. In these regards, our scheduling problem is hierarchical, where individual operations have their own scheduling problems. This problem arises because we wish to keep the system modular and scalable. In practice, the system hierarchy may not seem to be a major issue. However, it becomes problematic when considering error correction and control as discussed in the next section.

The final challenge results from the continuous nature of the scheduling problem. Currently jobs representing one or more designs are submitted to the robotic system in an asynchronous manner, effectively whenever anyone wants to build something. We currently schedule operations using a simple first-in/first-out queue where each job is run sequentially in batch mode. This mode of operation has not yet been problematic as we are still in the development phase, optimizing numerous experimental protocols for robotic operation. However, we are fast approaching the end of the development phase and hope soon to run the system in a continuous mode. When this occurs, the simple first-in/first-out policy will no longer suffice. We need to run multiple jobs concurrently and be able to add new jobs when existing ones are running. We also need to prioritize these jobs based on how urgently they are needed and how

easily they can integrate with existing jobs (e.g. container colocation and synchronization). The challenge, as first discussed above, is that each job is unique, requiring a unique sequence of operations, and that we continuously need to reschedule the ordering of operations on the robot as new jobs are added.

#### 4. ERROR CORRECTION AND CONTROL

Most designs result in failure. Failure commonly results from the design itself. As discussed above, there is still much uncertainty in the design process due to our limited knowledge of biology. For example, we may not know the enzyme necessary to perform a specific metabolic reaction or be unaware of some feedback loop that inhibits a key reaction. Trial and error is often the only solution and the principal justification for the cost in developing the robotic system. Equally common is failure during the assembly process. The individual steps required for DNA synthesis, assembly, and transformation often fail, and the efficiency of many steps is low, particularly as the design complexity increases. In some cases, these build failures result from using the wrong protocol, which can readily be identify during post-mortem analysis. However, in most other cases, they reflect the inherent stochasticity of many biochemical reactions. The solution here is often just to perform the step again, possibly using different conditions (e.g. temperature or concentrations) that will improve the overall efficiency of the reaction.

Often we can identify these build failures midway through the assembly process using measurement technologies such as spectroscopy or capillary/gel electrophoresis. Once identified, we can halt an existing process, identify the nature of the failure, and then possibly restart it from scratch or ideally some intermediate step. The challenge here is that restarting the job disrupts the scheduling of other jobs on the platform. Also, restarting from the end of the queue limits about ability to prioritize jobs and can add long lags in the assembly process, especially considering that these design failures are common. Dealing with these process disruption is a major challenge and will likely require entirely new approaches to formulating and solving the scheduling control problem.

#### 5. SUPPLY-CHAIN CONTROL AND DATA MANAGEMENT

Any schedule needs to account for material supplies. These include: tips for the pipettes; enzymes to perform different reactions; and media to grow cells. They also include DNA. We do not synthesize DNA ourselves but rather have external vendors synthesize it for us instead. Typically, we have short fragments of DNA synthesized (20-1000 base pairs in length) from these vendor and then assemble them into larger pieces on the robot. In a typical, low-complexity design for *S. cerevisiae*, for example, we need to assembly at least 10 of these smaller fragments into single larger one. Once assembled, we then need to transform it into cells.

Managing the supply chain is a major challenge. For one, there is the issue of storage. The platform has only a fixed amount of space for storing materials. In addition, some materials are labile, such as enzymes, cells, and DNA, and rapidly degrade at room temperature. While our system includes a refrigerator for storing these materials, space here is very limited and very expensive to expand (it is the most costly piece of equipment other than the platform itself and robotic liquid handler). Ultimately, some materials need to be stored off site. These materials can readily be added or removed when the robot is running. Determining which materials to store on or off the platform and when to change or replenish them is a key element of the control problem. Timing is also important. While the robot can run continuously, our graduate students and postdocs cannot. As a consequence, adding more tip boxes at noon is a far easier task than adding them at midnight. In addition, we need to factor in the time it takes to order and receive DNA and supplies from outside vendors. While the turnaround time is often a day or two, delays can arise due to bad weather or, in the case of DNA synthesis, build failures. Ideally, we can manage these inventories in real time, forecast needs, and adjust schedule when some materials become limiting. As all ordering is now done electronically, this process can be automated.

Data management is also an issue. We need to track samples and store all data associated with them. Some data are directly generated by the robot, such as logs and measurements made on the system. Others are generated off-site/off-line. For example, most our design are tested off-line. Often, we are trying to engineer microorganisms to produce chemicals or fuels. While in principle these fermentation could be performed on the robotic system, the small volumes required when using microplates are often impractical. We instead need to grow them at larger volumes in fermentors or shake flasks. In addition, we do not yet have these inline measurement capabilities on our system necessary to measure these chemical, specifically high-performance liquid chromatography or mass spectrometry. While we plan to include them in the future, our focus is currently on strain engineering as this is the major bottleneck in the design process.

We currently manage our data using a laboratory information system (LIMS). This has been relatively easy to develop and integrate with our robotic control system, at least with regards to data management, sample tracking, and inventory management. The basic system architecture is shown in Figure 4. The next step is to integrate the LIMS system with scheduling and supply chain management. The challenge here is not technical one, but nonetheless involves significant programming effort. It also reflects the complexity of the overall control problem.

## 6. OPPORTUNITIES

The process systems engineering community has traditionally focused on the industrial manufacturing sector, not surprisingly as this is where automation was first employed. However, automation, specifically through the use of robots, is now increasingly being employed in the laboratory for process development and discovery. The nature of these problems, however, is different than those in the manufacturing sector and introduce unique challenges as discussed above. In principle, these control problems can be solved using existing technologies. In fact, we ultimately



Fig. 4. System architecture.

envision that they will be solved by recasting the problem as a mixed-integer program embedded in a receding horizon (model predictive) control framework (e.g. Subramanian et al. (2012)). The challenge will be to adapt these technologies to deals with the specific problems associated with laboratory automation as detailed above.

The key hurdle, in our opinion, in adapting these technologies will be to develop a facile modeling framework that enables enables us to *unambiguously* formulate these control and scheduling problems, ideally in some modeling language. Currently, such as framework is lacking, and it is the reason why I discussed our numerous problems in such general terms. The most important step in solving any complex control problem is developing a good model, and currently we do not possess a reasonable strategy for generating them. Given such models, developing and evaluating different control strategies should become much easier. We believe this is a fertile area for future study.

The most significant opportunity is that our problem is real, and we and others are facing these challenges today. The robotic system is up and running. We are also not the only academic or industrial group currently or planning to to use robotics in synthetic biology. Nor is biology is not the only area where robotics are being employed. They are also being employed in chemical synthesis (Peplow, 2014; Li et al., 2015). Projecting forward, one can soon envision that all laboratories will eventually be automated to some degree. While cost is still a major barrier to implementation, the price point on the equipment side is rapidly decreasing. The real barriers to widespread implementation are the challenges associated with running these systems. Who better to solve these problems than the process systems community.

## 7. CONCLUSIONS

Synthetic biology is a new and rapidly emerging field of engineering with diverse applications in many fields. This paper has discussed many control problems arising from the use of robotics in synthetic biology. While the narrative draws heavily from the author's personal experiences at UIUC, robotic systems are increasingly being used in many academic and industrial laboratories. In addition, these systems being applied not only to solve problems in biology but also ones in chemistry. The goal is to motivate researchers from process systems engineering to develop solutions for these classes of scheduling and control problems.

#### ACKNOWLEDGEMENTS

The author thanks the members of the Biosystems Design Theme at the Institute of Genomic Biology (UIUC). Special thanks goes to members Huimin Zhao and Ran Chao, who were instrumental in designing the robotic system.

## REFERENCES

- Atsumi, S., Hanai, T., and Liao, J.C. (2008). Nonfermentative pathways for synthesis of branched-chain higher alcohols as biofuels. *Nature*, 451(7174), 86–89.
- Chen, S., Harrigan, P., Heineike, B., Stewart-Ornstein, J., and El-Samad, H. (2013). Building robust functionality in synthetic circuits using engineered feedback regulation. *Current Opinion in Biotechnology*, 24(4), 790–796.
- Clancy, K. and Voigt, C.A. (2010). Programming cells: towards an automated 'Genetic Compiler'. Current Opinion in Biotechnology., 21(4), 572–581.
- Gibson, D.G., Glass, J.I., Lartigue, C., Noskov, V.N., Chuang, R.Y., Algire, M.A., Benders, G.A., Montague, M.G., Ma, L., Moodie, M.M., Merryman, C., Vashee, S., Krishnakumar, R., Assad-Garcia, N., Andrews-Pfannkoch, C., Denisova, E.A., Young, L., Qi, Z.Q., Segall-Shapiro, T.H., Calvey, C.H., Parmar, P.P., Hutchison, C.A., Smith, H.O., and Venter, J.C. (2010). Creation of a bacterial cell controlled by a chemically synthesized genome. *Science*, 329(5987), 52–56.
- Gibson, D.G., Young, L., Chuang, R.Y., Venter, J.C., Hutchison, C.A., and Smith, H.O. (2009). Enzymatic assembly of DNA molecules up to several hundred kilobases. *Nature Methods*, 6(5), 343–345.
- Lee, J.W., Na, D., Park, J.M., Lee, J., Choi, S., and Lee, S.Y. (2012). Systems metabolic engineering of microorganisms for natural and non-natural chemicals. *Nature Chemical Biology*, 8(6), 536–546.
- Li, J., Ballmer, S.G., Gillis, E.P., Fujii, S., Schmidt, M.J., Palazzolo, A.M., Lehmann, J.W., Morehouse, G.F., and Burke, M.D. (2015). Synthesis of many different types of organic small molecules using one automated process. *Science*, 347(6227), 1221–1226.
- Moon, T.S., Lou, C., Tamsir, A., Stanton, B.C., and Voigt, C.A. (2012). Genetic programs constructed from layered logic gates in single cells. *Nature*, 491(7423), 249–253.
- Peplow, M. (2014). Organic synthesis: The robo-chemist. Nature, 512(7512), 20–22.
- Rao, C.V. and Arkin, A.P. (2001). Control motifs for intracellular regulatory networks. Annual Reviews in Biomedical Engineering, 3, 391–419.
- Ro, D.K., Paradise, E.M., Ouellet, M., Fisher, K.J., Newman, K.L., Ndungu, J.M., Ho, K.A., Eachus, R.A., Ham, T.S., Kirby, J., Chang, M.C., Withers, S.T., Shiba, Y., Sarpong, R., and Keasling, J.D. (2006). Production of the antimalarial drug precursor artemisinic acid in engineered yeast. *Nature*, 440(7086), 940–943.
- Schirmer, A., Rude, M.A., Li, X., Popova, E., and del Cardayre, S.B. (2010). Microbial biosynthesis of alkanes. *Science*, 329(5991), 559–562.

- Subramanian, K., Maravelias, C.T., and Rawlings, J.B. (2012). A state-space model for chemical production scheduling. *Computers & Chemical Engineering*, 47, 97 – 110.
- Way, J.C., Collins, J.J., Keasling, J.D., and Silver, P.A. (2014). Integrating biological redesign: where synthetic biology came from and where it needs to go. *Cell*, 157(1), 151–161.
- Wu, K. and Rao, C.V. (2010). The role of configuration and coupling in autoregulatory gene circuits. *Molecular Microbiology*, 75(2), 513–527.
- Xue, Z., Sharpe, P.L., Hong, S.P., Yadav, N.S., Xie, D., Short, D.R., Damude, H.G., Rupert, R.A., Seip, J.E., Wang, J., Pollak, D.W., Bostick, M.W., Bosak, M.D., Macool, D.J., Hollerbach, D.H., Zhang, H., Arcilla, D.M., Bledsoe, S.A., Croker, K., McCord, E.F., Tyreus, B.D., Jackson, E.N., and Zhu, Q. (2013). Production of omega-3 eicosapentaenoic acid by metabolic engineering of *Yarrowia lipolytica*. *Nature Biotechnology*, 31(8), 734– 740.

# Copyright © 2015 IFAC