Chapter 15

Genome-Scale Metabolic Network Reconstruction

Marco Fondi and Pietro Liò

Abstract

Bacterial metabolism is an important source of novel products/processes for everyday life and strong efforts are being undertaken to discover and exploit new usable substances of microbial origin. Computational modeling and in silico simulations are powerful tools in this context since they allow the exploration and a deeper understanding of bacterial metabolic circuits. Many approaches exist to quantitatively simulate chemical reaction fluxes within the whole microbial metabolism and, regardless of the technique of choice, metabolic model reconstruction is the first step in every modeling pipeline. Reconstructing a metabolic network consists in drafting the list of the biochemical reactions that an organism can carry out together with information on cellular boundaries, a biomass assembly reaction, and exchange fluxes with the external environment. Building up models able to represent the different functional cellular states is universally recognized as a tricky task that requires intensive manual effort and much additional information besides genome sequence. In this chapter we present a general protocol for metabolic reconstruction in bacteria and the main challenges encountered during this process.

Key words Metabolic model reconstruction, Flux balance analysis, Metabolic modeling

1 Introduction

One of the most important drawbacks derived from the booming of genomics resides in the possibility to (almost) automatically derive the potential metabolic landscape of a strain, given its genome. This is of particular importance when dealing with biotechnologically or clinically relevant strains since metabolism represents a key factor for understanding their physiology. In general, living organisms possess complex metabolic networks, ranging from hundreds to thousands of chemical reactions and conferring them the capability to synthesize and/or catabolize the building blocks of their cells. The sum of these chemical reactions represents the core of any living organism and the coordination of these processes results in the physiology we associate to each organism, from bacteria to humans [1]. Bacteria, in particular, continuously provide industry with novel products/processes based on the use of their metabolism and numerous efforts are being undertaken

Alessio Mengoni et al. (eds.), *Bacterial Pangenomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1231, DOI 10.1007/978-1-4939-1720-4_15,

worldwide, with an ultimate goal to deliver new usable substances of microbial origin to the marketplace [2], including pharmaceuticals, biofuels, and bioactive compounds in general. Classical examples of industrial bio-based production of valuable compounds include vitamin C [3], xanthan (E425) [4], isopropanol, butanol and ethanol mixture [5], and succinate [6].

The importance of bio-based products in everyday life has tremendously boosted research on microbial metabolic processes and understanding the basic functioning of the biosynthetic circuits of living cells has become a crucial issue in systems microbiology. In this context, computational modeling and in silico simulations are often adopted by metabolic engineers to quantitatively simulate chemical reaction fluxes within the whole microbial metabolism [7, 8]. Among possible approaches, the so-called constraint-based methods (e.g., flux balance analysis, FBA, [9]) can be applied to large (genome-scale) biochemical systems since they require only the information on metabolic reaction stoichiometry and mass balances around the metabolites under pseudo-steady-state assumption [10]. Thus, according to this methodology, detailed information on the chemical equations of the studied system is not required. Genome-scale metabolic modeling has become an important tool in the study of metabolic networks in pathogens [11], and chemical [3] and environmental [11] research areas. Methods and tools for in silico metabolic modeling have been recently reviewed in [12], [13], and [14, 15], respectively.

To exploit computational approaches, cellular metabolic networks are transformed into a model by drafting the list of the biochemical reactions that an organism can carry out together with the boundaries of the system, a biomass assembly reaction, and exchange fluxes with the environment [16]. These reconstructions account for the functions of hundreds to thousands of genes, and are ideally intended to incorporate all known metabolic reactions for a particular organism into a standardized format, enabling the generation of a computational model that can be analyzed with a variety of emerging mathematical techniques [8]. Constraint-based modeling framework can be used to automatically compute the resulting balance of all the chemical reactions predicted to be active in the cell and, in turn, to bridge the gap between knowledge of the metabolic network structure and observed metabolic phenotypes.

The process of reconstructing and validating a metabolic model is a complex task. Currently, about 4,000 complete genome sequences are available in public databases (www.genomesonline.org); conversely, only around 100 reconstructions of microbial metabolic systems can be retrieved (see http://systemsbiology.ucsd.edu/InSilicoOrganisms/OtherOrganisms for an updated list). This gap is the most evident consequence of the difficulties in

reconstructing "working" metabolic models starting from genome annotations and is a key challenge for future systems microbiology.

Drafting a metabolic model of an organism nowadays is almost straightforward since many tools able to make this step automatic are available [17–21] (described in details below); however, turning these reconstructions into models capable of fully representing the functional states of a given organisms is not trivial. In particular, most of the draft metabolic available to date are incomplete because (1) they often do not include essential metabolic steps for sustaining in silico cellular growth (metabolic gaps) and (2) key issues of embedded reactions such as stoichiometry, directionality, and charge are sometimes missing or erroneous. Moreover, since draft models are mainly reconstructed on the basis of sequence homology in respect to other (closely related) microorganisms, they will not include organism-specific metabolic pathways (often responsible for key phenotypic features). To overcome these difficulties and guide model revision, Thiele and Palsson [22, 23] have built a protocol including (at least) 4 stages and 94 different steps necessary for the reconstruction of reliable, high-quality, metabolic models. Importantly, a large fraction of these steps cannot be performed in an automated fashion, thus requiring intense and time-consuming manual effort/curation. Speeding up some of the steps represents one of the most important achievements required for accelerating the overall process of metabolic modeling and engineering of microbial strains.

This chapter is intended as a general protocol for bacterial metabolic model reconstruction and will describe the main steps encountered during this process (using FBA as the modeling framework). Further details and specific challenges of each step herein described can be found in previous (and more detailed) papers [22–24].

2 Materials

In this section we describe what is needed for starting to reconstruct the metabolic model of a strain under study.

2.1 Genome Sequence

Genome sequence is nowadays the most widely adopted resource for drafting the metabolic model of an organism. So, to start the reconstruction you will need a FASTA file embedding a set of contigs or coding sequences of the genome you want to analyze.

2.2 Online Reconstruction Tools

A number of tools exist for drafting the metabolic model of a given organism (Table 1).

1. The coupling of *RAST* and *Model SEED* pipelines provides a fully automated annotation and model reconstruction service

Table 1
List of software/methods for automatic metabolic reconstructions

Name	Reference	Website	Standalone version	Free (F)/ commercial (C)
RAST/Model Seed	[17, 25, 50]	http://rast.nmpdr.org/, http://www.theseed.org/	Not available	F
MicrobesFlux	[19]	http://tanglab.engineering.wustl.edu/static/MicrobesFlux.html	Not available	F
FAME	[51]	http://f-a-m-e.org/	Not available	F
Pathway Tools Software	[20, 28]	http://bioinformatics.ai.sri.com/ptools/	Available	F/C
COPABI	[21]	–	Not available	F
CARMEN	[18]	http://carmen.cebitec.uni-bielefeld.de		F
Kbase	–	www.kbase.us	Available	F
GEMSiRV	[34]	http://sb.nhri.org.tw/GEMSiRV/en/GEMSiRV	Available	F
RAVEN	[35]	http://www.sysbio.se/BioMet	Available	F
Metashark	[33]	http://bioinformatics.leeds.ac.uk/shark/	Available	F
SuBliMinaL Toolbox	[36]	http://www.mcisb.org/resources/subliminal/	Available	F

for archaeal and bacterial genomes. The service seeks to rapidly produce high-quality assessments of gene functions and, most importantly in the context of the present chapter, an initial (draft) metabolic reconstruction [17]. Each preliminary model network includes all reactions associated with one or more enzymes encoded in the organism's genome as well as a set of spontaneous reactions that do not require enzymatic catalysis [17, 25]. Importantly, Model SEED also provides tools for preliminary analysis of reconstructed metabolic networks, including auto-filling of metabolic gaps and FBA of the model. Overall, about ~48 h is necessary to reconstruct a metabolic model from an assembled genome sequence.

2. *MicrobesFlux* is a platform to build metabolic models for all the organisms whose completely sequenced genome is present in the KEGG (Kyoto Encyclopedia of Genes and Genomes) database. Indeed, this tool is able to automatically download the metabolic network (including enzymatic reactions and metabolites) of ~1,200 species from the KEGG database and then convert it to a metabolic model draft.

3. The Flux Analysis and Modeling Environment (*FAME*) is a web-based modeling tool that allows the reconstruction of metabolic models. In addition this tool also includes other tasks, such as editing, running, and analyzing/visualizing stoichiometric models. FAME allows users to either upload their own preexisting model or to build a new model. To perform this latter task, this software takes advantage of the metabolic models stored in the KEGG database [26]. Importantly, FAME is quite flexible, allowing any stoichiometric model to be loaded into FAME, provided it is encoded in the Systems Biology Markup Language (SMBL, see below).
4. Similarly, the software tool *CARMEN* [18] performs in silico reconstruction of metabolic networks to help translate genomic data into functional ones. CARMEN also enables the visualization of automatically derived metabolic networks based on pathway information from the KEGG database [26] or from user-defined SBML templates.
5. *Pathway Tools* [20, 27, 28] is a software environment for management, analysis, and visualization of integrated collections of genome, pathway, and regulatory data. This tool can be used for de novo genome-scale model generation and also for other post-processing tasks such as interactive editing, visualization, and comparative analyses. Recently, PathwayTools has been used for a systematic comparison between KEGG and MetaCyc [29, 30] databases, revealing differences in the two repositories in that KEGG contains significantly more compounds than does MetaCyc, whereas MetaCyc contains more reactions and pathways than does KEGG; in particular KEGG modules are quite incomplete [31].
6. A Computational Platform for the Access of Biological Information (*COPABI*) has been recently developed by Reyes et al. [21]. This platform allows the automation of a methodology for the reconstruction of genome-scale metabolic models for any organism. The algorithm comprises several steps including (1) the information compilation from free-access biological databases, (2) interaction of the user with the platform in order to properly select the parameters for the probabilistic criteria and choices for the biomass components and restrictions, and finally (3) application of unicity and completeness criteria and production of the output. Unicity criterion aims at identifying reactions that appear more than once and also identifies their enzymes. Repeated reactions are then eliminated following the criterion according to which the enzyme that appears less frequently in the model is not eliminated. Completeness aims at adding novel reactions to the model in order to fill the gaps that are commonly found in the draft reconstruction process.

7. The metabolic SearcH And Reconstruction Kit (*metaSHARK*) [32, 33] is a new fully automated software package for the detection of enzyme-encoding genes within unannotated genome data and their visualization in the context of the surrounding metabolic network. Unlike most of the previously described reconstruction tools that start with a set of predicted proteins from an annotated genome and that, by a variety of text mining and/or sequence comparison methods, construct a list of the enzymatic functions that are asserted to be present, metaSHARK only requires a set of DNA sequences [finished chromosomes, contigs, genome survey sequences, or expressed sequence tags (ESTs)] as input, and hence can be applied to extract new knowledge of metabolic capabilities from preliminary data produced by unannotated and ongoing genome sequencing projects.
8. The recently proposed *KBase* (http://kbase.science.energy.gov/) is a software environment designed to enable researchers to reconstruct, optimize, and analyze genome-scale metabolic models. Genome-scale metabolic models are reconstructed starting from an annotated genome object using the DOE Systems Biology Knowledgebase tools.
9. The metabolic network reconstruction module of *GEMSIRV* (GEnome-scale Metabolic model Simulation, Reconstruction and Visualization) [34] allows editing/updating the content of a model that has been previously imported (in SBML or spreadsheet format), using other models of closely related species as a guide. Alternatively, a draft reconstruction can be generated by mapping a blank reconstruction (containing gene information only) to a reference reconstruction.
10. *Raven* (Reconstruction, Analysis, and Visualization of Metabolic Networks) [35] is a tool for automatic reconstruction of GEMs based on protein orthology and (optionally) on a set of already available genome-scale metabolic models. The method takes advantage of the KEGG Orthology (KO) IDs for inferring gene-protein-reaction association.
11. *SuBliMinaL* toolbox [36] is a collection of methods enabling the automated reconstruction of genome-scale metabolic models, exploiting both KEGG and MetaCyc resources. In the generated model, all the metabolic pathways described in each resource are merged and can be used for the successive pipeline step (annotation), in which already existing reconstructions can be used for improving the de novo-reconstructed model.

2.3 SBML-Formatted Metabolic Model

SBML (Systems Biology Markup Language) is a software-independent language for describing different biological processes and is nowadays considered the standard medium for representation

and exchange of biochemical network models [37]. In its general formulation, it resembles the basic features of the XML data stream [38] and allows representing all the elements accounting for biochemical reactions, including (1) the cellular compartment(s) in which the reaction occurs, (2) chemical species involved (substrates and products), (3) the reversibility (or irreversibility) of each reaction, and (4) unit definition (according to which quantities of substrates and/or products that are consumed and/or produced are expressed). A simple model (seven compounds, one reaction) together with its SBML counterpart is reported in Fig. 1. This model represents the ATP-dependent transport of (periplasmic) D-glucose into the cellular cytoplasm according to the iAF1260 metabolic reconstruction of *Escherichia coli* [39]. As shown in Fig. 1b, this SBML representation can be divided into three main sections: the first part (black font) includes general details on the reconstructed model such as the organism (*E. coli* iAF1260), the unit definition (mmol/gDW h), and the model compartments (extracellular, periplasm, cytosol). The second section (blue font) includes the list of the chemical species that the model will be able to recognize and handle (together with their name, formula, charge and boundary condition). The last section of this small model (red font) lists all the possible biochemical transformations

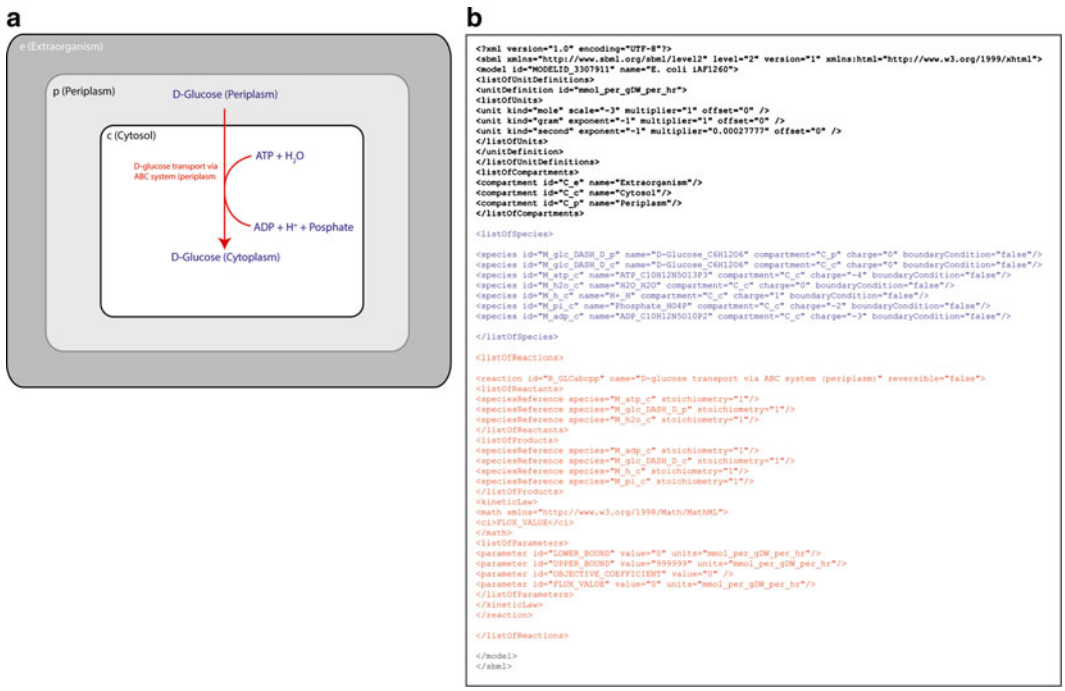

Fig. 1 Schematic representation of a one-reaction metabolic model in a compartmentalized cell (**a**) and the corresponding SMBL code (**b**)

(only "D-glucose transport via ABC system" in this case) together with information on the reversibility of the reactions, the stoichiometric coefficients of substrates/products, and reaction bounds (upper and lower, UB and LB, respectively).

2.4 Modeling Framework

There are several available tools for performing constraint-based metabolic modeling (reviewed in [24]). SBML-formatted models are generally recognized by these tools and can be imported/converted for successive computation. Among them, COBRA toolbox [40] is probably the most widely adopted. The original version of this package is to be used within the Matlab (The Mathworks Inc.) numerical computation and visualization environment although, recently, a version exploiting Python programming language has been developed (COBRApy, [41]). Command lines reported in this protocol refer to the Matlab-based version of COBRA toolbox. When available, their COBRApy counterpart is also reported. Also needed are libSMBL (an API library for manipulation of systems biology models) [42] and a Linear Programming (LP) solver supported by the COBRA Toolbox as, for example, gkpl (http://www.gnu.org/software/glpk) or Gurobi (Gurobi Optimization, http://www.gurobi.com). Please refer to specific literature/manuals/websites for information on the installation and configuration of these tools.

3 Methods

3.1 Obtain a Draft Metabolic Model

Most of the tools in Table 1 allow uploading a draft genome (or a set of coding sequences, CDS) and return an SBML-formatted metabolic model. RAST, for example, can generate a draft metabolic model just by selecting the "*Build metabolic model*" option before starting genome annotation process. MicrobesFlux allows creating metabolic models from all organisms present in KEGG database and extract them in SBML format by clicking the "*Get SBML*". With KBase one can generate an SBML-formatted model either selecting a genome that is already in the KBase Central Data Store (CDS) or a genome that has been already annotated; metabolic models are reconstructed through the function "*genomeTO_to_reconstructionTO*".

Regardless of the tool used, the output of this preliminary step is a draft SBML metabolic reconstruction (Fig. 2) that still needs manual curation to be turned into a functional model.

3.2 Model Evaluation

3.2.1 Missing Reactions and Alternative Pathways

At this stage, the reconstructed model may be incomplete and lack metabolic genes and/or functions. Thus, before starting modeling procedures, it is important to check possible sources of errors. To do this, revise the reconstructed metabolic model in a pathway-by-pathway manner to highlight, for example, potential missing

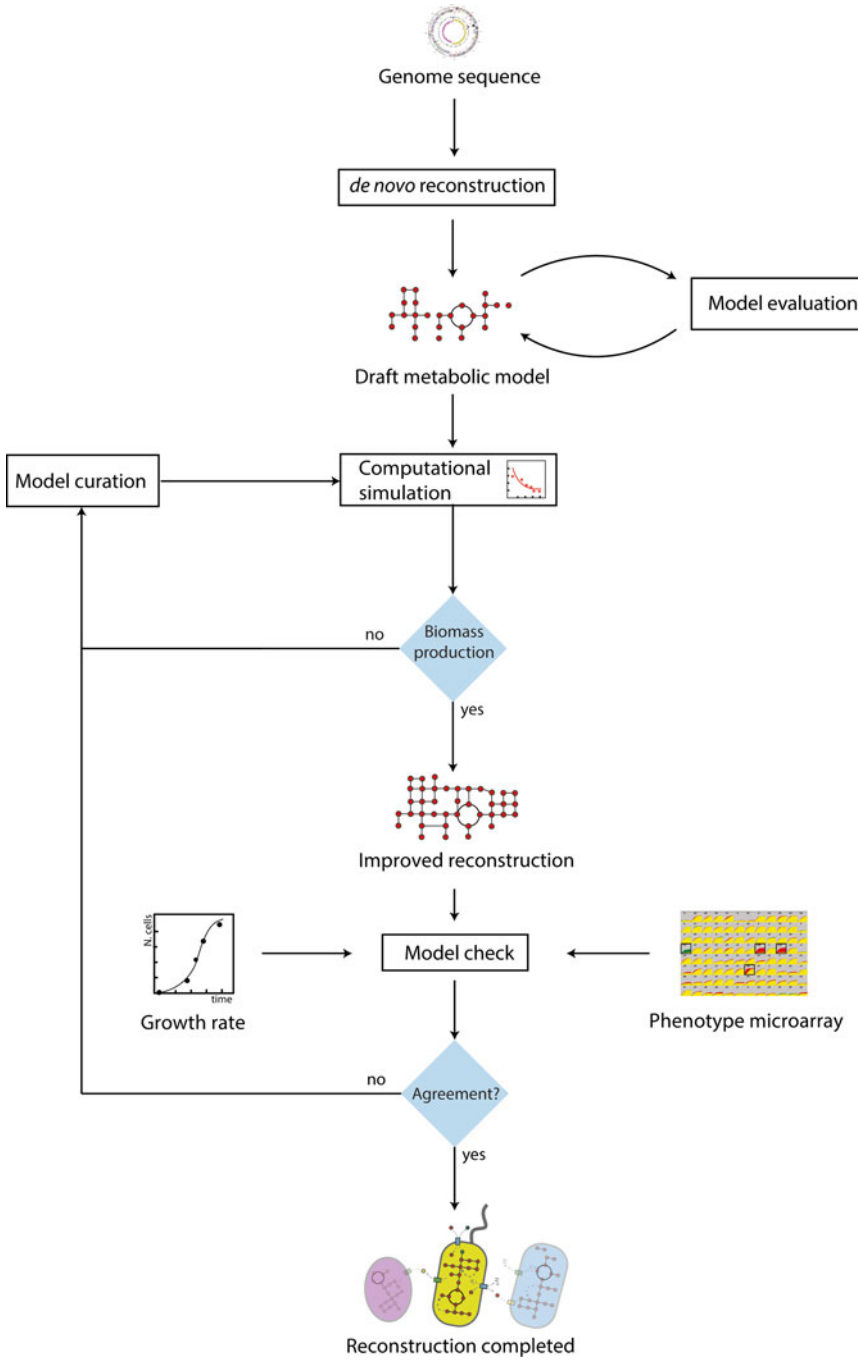

Fig. 2 Schematic representation of a pipeline for metabolic model reconstruction and checking

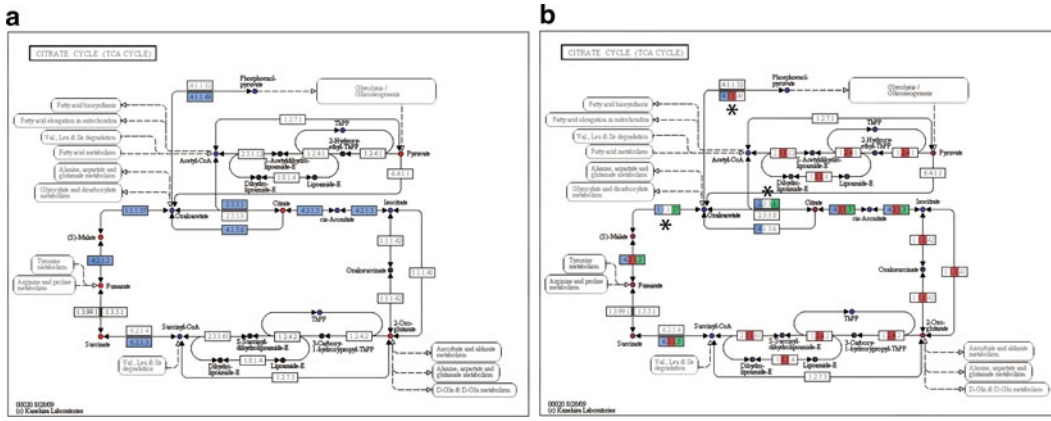

Fig. 3 Reconstructions of the citrate cycle from three Gamma-proteobacteria have been compared [*E. coli* K12 (*red boxes*), *P. haloplanktis* TAC125 (*blue boxes*) and *A. baylyi* ADP1 (*green boxes*)]. Reactions marked with "*" represent paradigmatic examples of how comparative genomics can be exploited for gap-filling metabolic models

reactions. To accomplish this task, graphical visualization of metabolic pathways is highly recommended. On the SEED user page one can display the different metabolic maps for the model under study and browse them just by clicking on the biosynthetic route name. In Fig. 3a citrate cycle is shown for the iAF1260 reconstruction of *E. coli*. As shown in this figure, reactions are colored according to their presence within the studied reconstruction. By doing so information on each metabolic step can be obtained just by clicking on the reaction E.C. code and possible gaps (interrupted pathways) and/or alternative metabolic steps can be easily identified. Record every potential missing reaction or any other unusual metabolic reaction in a spreadsheet and store as much information as possible [e.g., enzyme E.C. number, reaction code (starting with *rxn* in SEED model viewer), metabolic pathway] for each of them. Once all the pathways have been examined, this list should be carefully analyzed and integrated with as much information as possible. In particular, in case of potentially missing reactions one should:

1. *Check available scientific literature and metabolic databases* for the microbe under study since alternative metabolic steps may have been previously described for a given pathway. Also, data on metabolic auxotrophies of the strain of interest may be of interest (e.g., results from Biolog Phenotype Microarray experiments) in this phase.
2. *Use comparative genomics of closely related microorganisms.* SEED model viewer is of great help in this stage since multiple reconstructions can be simultaneously displayed over the same map. To do this, select other models of closely related organisms

from the initial (log-in) page and then click on the biosynthetic pathway you want to examine in detail. A possible output of this procedure is shown in Fig. 3b, in which reconstructions of the citrate cycle from three Gamma-proteobacteria have been compared (*E. coli* K12, *Pseudoalteromonas haloplanktis* TAC125 and *Acinetobacter baylyi* ADP1). Reactions marked with "*" represent paradigmatic examples of how comparative genomics can be exploited for gap-filling metabolic models. Indeed, the presence of those reactions in two organisms out of three might suggest potential errors during the metabolic reconstruction of the other strain. To validate this indication retrieve the sequence of the enzyme encoding for that reaction in one of the organisms possessing it and perform a BLAST search in the genome of the organism missing it. The presence of a hortologous sequence in the probed genome is a strong indication for gap filling the corresponding metabolic step. Other databases can be explored for retrieving information on the metabolic features of closely related microorganisms including KEGG [26] and MetaCyc [43].

The possibility to introduce a confidence score for each reaction added in this stage has been proposed [22], accounting for the type of evidence used for including the reactions within the reconstruction and ranging from 5 (in case biochemical data is available for that specific step) to 1 (in case that reaction has been included only for modeling purposes and no experimental evidence has been provided). These codes are particularly useful during model curation since low-confidence reactions can easily be identified.

3.2.2 Check Reaction Consistency

Each reaction present in the metabolic model at this stage should be carefully inspected in order to check (at least):

1. Substrate and cofactor usage
2. Charged formula for each metabolite
3. Reaction stoichiometry
4. Reaction directionality
5. Information for gene and reaction localization
6. Gene-protein-reaction (GPR) associations

See [22] for detailed instruction on how to accomplish each of these sub-steps.

3.3 Define Biomass Reaction

Reconstructed models usually embed an "artificial" reaction accounting for the assembly of all known biomass components (e.g., DNA, RNA, lipids, proteins, peptidoglycan) and their relative contributions to the overall cellular biomass. As an example, the biomass assembly reaction from the iAbaylyiV4 reconstruction of *A. baylyi* ADP1 is expressed as follows (according to Model SEED, *see* **Note 1** for compound name):

```
<reaction  id="rxn12832"  name="Biomass assembly"  reversible="false">
<notes>
<html:p>GENE_ASSOCIATION:UNKNOWN</html:p>
</notes>
<listOfReactants>
<speciesReference  species="cpd00001_c"  stoichiometry="40"/>
<speciesReference  species="cpd00002_c"  stoichiometry="40"/>
<speciesReference  species="cpd11461_c"  stoichiometry="0.032"/>
<speciesReference  species="cpd11462_c"  stoichiometry="0.2"/>
<speciesReference  species="cpd11463_c"  stoichiometry="0.633"/>
<speciesReference  species="cpd11649_c"  stoichiometry="0.003"/>
<speciesReference  species="cpd11677_c"  stoichiometry="0.002"/>
<speciesReference  species="cpd16601_c"  stoichiometry="0.002"/>
<speciesReference  species="cpd16653_c"  stoichiometry="0.032"/>
<speciesReference  species="cpd16661_e"  stoichiometry="0.028"/>
<speciesReference  species="cpd16662_c"  stoichiometry="0.041"/>
<speciesReference  species="cpd16663_c"  stoichiometry="0.021"/>
<speciesReference  species="cpd16669_c"  stoichiometry="0.006"/>
</listOfReactants>
<listOfProducts>
<speciesReference  species="cpd00008_c"  stoichiometry="40"/>
<speciesReference  species="cpd00009_c"  stoichiometry="40"/>
<speciesReference  species="cpd11416_c"  stoichiometry="1"/>
</listOfProducts>
<kineticLaw>
       <math xmlns="http://www.w3.org/1998/Math/MathML">
                    <ci> FLUX_VALUE </ci>
       </math>
       <listOfParameters>
              <parameter id="LOWER_BOUND" value="0" name="mmol_per_gDW_per_hr"/>
              <parameter id="UPPER_BOUND" value="10000" name="mmol_per_gDW_per_hr"/>
              <parameter id="OBJECTIVE_COEFFICIENT" value="0.0"/>
              <parameter id="FLUX_VALUE" value="0.0" name="mmol_per_gDW_per_hr"/>
       </listOfParameters>
</kineticLaw>
</reaction>
```

Conventionally, the biomass reaction is expressed in h^{-1}, since precursor fractions are converted to mmol/gDW. The biomass assembly reaction sums the mole fraction of each precursor necessary to produce 1 g dry weight of cells [22].

So, at this point, scan available literature for biomass composition of the strain under study. In case available experimental data is not enough, you may derive missing pieces of information from the biomass composition from (more studied) closely related strains. Store information on biomass components in a spreadsheet and then add the assembly reaction into the draft model.

As shown for iAbaylyiV4 reconstruction, biomass assembly reaction should also account for the energy (in the form of ATP) necessary for cell replication. This is usually referred to as GAM (growth-associated ATP maintenance) reaction and can be calculated experimentally. In case no experimental information is available, one can approximate growth-associated costs from the GAM reaction of a closely related strain or deriving it from an estimation of total amount of ATP required to synthesize cellular macromolecules (protein, DNA, and RNA) whose amount can be derived from databases. This latter step is fully described in [22].

3.4 Define Additional Exchange Reactions

Exchange reactions (conventionally labeled with "EX_") allow defining the composition of the in silico growth medium and environmental conditions during simulations. In other words, these reactions define the range of compounds that can be imported into the cellular model and metabolized to form biomass constituents or other cellular products. At this stage, the draft model should already include a minimal set of exchange reactions. As an example, a typical exchange reaction allowing the model to use glucose (ModelSEED code `cpd00027`) can be represented as follows:

```
<reaction  id="EX_cpd00027_e"  name="EX_D-Glucose_e"  reversible="true">
        <notes>
                <html:p>GENE_ASSOCIATION: </html:p>
                <html:p>PROTEIN_ASSOCIATION: </html:p>
                <html:p>SUBSYSTEM: S_</html:p>
                <html:p>PROTEIN_CLASS: </html:p>
        </notes>
        <listOfReactants>
                <speciesReference  species="cpd00027_e" stoichiometry="1.000000"/>
        </listOfReactants>
        <listOfProducts>
                <speciesReference  species="cpd00027_b" stoichiometry="1.000000"/>
        </listOfProducts>
        <kineticLaw>
                <math xmlns="http://www.w3.org/1998/Math/MathML">
                                <ci> FLUX_VALUE </ci>
                </math>
                <listOfParameters>
                        <parameter id="LOWER_BOUND" value="-10000"
units="mmol_per_gDW_per_hr"/>
                        <parameter id="UPPER_BOUND" value="10000"
units="mmol_per_gDW_per_hr"/>
                        <parameter id="OBJECTIVE_COEFFICIENT" value="0.0"/>
                        <parameter id="FLUX_VALUE" value="0.000000"
units="mmol_per_gDW_per_hr"/>
                </listOfParameters>
        </kineticLaw>
</reaction>
```

This particular reaction allows transforming `cpd00027_b` into `cpd00027_e` that will be then used by the other reactions of the model (for example by a transport reaction that will convert `cpd00027_e` into its cytoplasmic counterpart `cpd00027_c`).

So, in this step, check every exchange reaction in respect to specific growth requirements of the strain under study. Information on commonly growth media is highly useful in this phase. Check the composition of every known growth medium for the strain under analysis and include one exchange reaction for each constituent of the growth medium. These reactions can also be added afterwards as they will be used in the next steps to predict cellular growth in specific nutritional conditions.

3.5 Validation of the SBML Model

Validate your model and search for potential formatting errors using the online tool SBML Validator at http://sbml.org/Facilities/Validator/. If no errors are issued it means that your reconstruction is a valid SBML model and you can use it for the next step.

3.6 Import Model into Modeling Framework

The reconstruction is now ready for being imported into COBRA toolbox and start metabolic modeling procedures using FBA. We will assume that Matlab, COBRA toolbox, libSBML, and a valid LP solver (together with their dependencies) have been successfully installed and initialized on the workstation. Assuming that the SBML-formatted model is stored in a file called *draft_model.xml*, import the reconstruction into Matlab with

```
model = readCbModel('draft_model.xml')
```

In Cobrapy run `model=create_cobra_model_from_sbml('draft_model.xml')`

If the model has been correctly imported, you should see something similar to this (within the Matlab console):

```
model =
                rxns: {1284x1 cell}
                mets: {1108x1 cell}
                   S: [1108x1284 double]
                 rev: [1284x1 double]
                  lb: [1284x1 double]
                  ub: [1284x1 double]
                   c: [1284x1 double]
          metCharge: [1108x1 int32]
               rules: {713x1 cell}
               genes: {713x1 cell}
         rxnGeneMat: [1284x713 double]
            grRules: {1284x1 cell}
         subSystems: {1284x1 cell}
    confidenceScores: {1284x1 cell}
       rxnReferences: {1284x1 cell}
        rxnECNumbers: {1284x1 cell}
            rxnNotes: {1284x1 cell}
            rxnNames: {1284x1 cell}
            metNames: {1108x1 cell}
         metFormulas: {1108x1 cell}
          metChEBIID: {1108x1 cell}
           metKEGGID: {1108x1 cell}
        metPubChemID: {1108x1 cell}
      metInChIString: {1108x1 cell}
                   b: [1108x1 double]
         description: 'draft_model.xml'
```

3.7 Check for Model Consistency

COBRA toolbox allows evaluating the imported reconstruction in a global fashion. So, before starting modeling procedures use it to check for:

1. Mass, charge, and stoichiometrically unbalanced reactions: For this you can use the COBRA function:

   ```
   [massImbalance,imBalancedMass,imBalancedCharge,imBalancedBool,Elements] =
   checkMassChargeBalance(model)
   ```

   (In Cobrapy: `unbalanced_rxns=[r for r in model.reactions if r.check_mass_balance() != []]`)

   Fix reactions listed in `imBalancedMass` and `imBalancedCharge` adding, for example, missing protons or proton donors.

2. Gaps in reconstruction (dead-end metabolites, i.e., metabolites that are produced but not consumed): For this you can use the COBRA function:

```
[allGaps,rootGaps,downstreamGaps] = gapFind
(model)
```

Fill gaps in `allGaps` by searching for possible reactions involved in the consumption/production of identified dead-end metabolites. You may use the same approach described in Subheading 3.2.1.

3.8 Set In Silico Medium Composition

The composition of the growth medium can be defined tuning lower and upper bounds (LB and UB, respectively) of exchange reactions in the model (*see* Subheading 3.4). Indeed, through LB and UB it is possible to define the maximum utilization rate for each of the compounds to be imported into the model through exchange reactions. Conventionally, uptake (utilization) rates for a given compound are defined by tuning LB values of the corresponding exchange reaction. Water and inorganic ions are usually considered to be present in non-limiting concentrations and LBs of their corresponding exchange reactions are set to very high values (e.g., 1,000 mmol/g h). Conversely, setting the LB of the exchange reaction regulating the utilization rate of the carbon source(s) present in the medium requires much more attention. Defining a wrong value here would reveal in unreal prediction of cellular growth rate (*see* **Note 2** for details on uptake ration calculation). The LBs of all the other exchange reactions present in the model must be set to "0".

Hence, according to the growth medium in which you want to test the model:

1. For each of the exchange reactions of inorganic ions (in this case `cpd00048`, sulfate) present in the growth medium, set LB to "1,000 mmol/g h", using the `changeRxnBounds` COBRA toolbox function:

```
model = changeRxnBounds(model, 'EX_cpd00048
(e)', -1000, 'l')
```

In Cobrapy run

```
rxn= model.reactions.get_by_id(EX_cpd00048(e))
rxn.lower_bound=-1000.0
rxn.upper_bound=1000.0
```

where `EX_cpd00048(e)` is the exchange reaction for sulfate. Alternatively, you can first define a Matlab list embedding all the exchange reactions of inorganic ions present in the model with

```
IonExchangeReactions={'EX_cpd00048(e)', 'EX_
cpd00067(e)', …}
```

and then set their LB value with

```
model = changeRxnBounds(model, IonExchange
Reactions, -1000, 'l')
```

2. Set the LB of the exchange reaction for the carbon source to some realistic value. In this example we will use glucose (`'cpd00027'`) as the carbon source and we will set the LB of the corresponding exchange reaction to 18 mmol/g h [a value that has been calculated for *E. coli* during fed batch growth [7]]:

```
model = changeRxnBounds(model, 'EX_cpd00048
(e)', -18, 'l')
```

3. Set the LBs of all the remaining exchange reactions to "0". We assume that these reactions have been stored in a list called `RemainingEXreactions`:

```
model = changeRxnBounds(model, Remaining
EXreactions, 0, 'l')
```

3.9 Optimize Model for Biomass Production

The model is now ready for optimization. First of all, identify one reaction of the model as the optimization objective function, i.e., the reaction of the model you want to maximize during simulations. By doing this, linear programming can be used to infer the flux distribution that maximizes (or minimizes) the output of that specific reaction. At this stage of the reconstruction, biomass production should be set as the model objective function. In this way, one can test whether all the compounds involved in biomass assembly (*see* Subheading 3.3) can be synthesized or not. In the first case, the flux out of the biomass assembly (`f`) will be greater than 0; conversely, in case one (or more) biomass constituent(s) cannot be produced, `f` will be equal to 0. Use the following command to define the objective function of the model with COBRA:

```
model = changeObjective(model, 'rxn12832')
```

In Cobrapy run

```
rxn=model.reactions.get_by_id('rxn12832')
rxn.objective_coefficient = 1.0
```

where `rxn12832` is the biomass assembly reaction as defined by the model in this specific case. Then, to derive the flux distribution that optimizes the flux through objective reaction (exploiting FBA), use the following command:

```
FBAsolution = optimizeCbModel(model,'max')
```

In Cobrapy run

```
model.optimize(solver='gurobi')
print model.solution)
```

In a Matlab console, the output of this command should look like this:

```
FBAsolution =
           x: [1284x1 double]
           f: 0
           y: [1108x1 double]
           w: []
        stat: 1
    origStat: -99
      solver: 'gurobi5'
        time: 0.0131
```

In this example, the value of `FBAsolution.f` is 0. This means that one (or more) substrate(s) of the biomass assembly reaction cannot be produced, most likely because of missing reactions (gaps) in the model.

3.10 Manual Curation

To identify which of the biomass precursor(s) cannot be synthesized, repeat the following points for each of them:

1. Add an artificial exchange reaction to the model, accounting for the extrusion of that compound, as shown here for compound `cpd00155_c` of the biomass assembly reaction of Subheading 3.3:

```
<reaction  id="EX_ cpd00155_e"  name="EX_Glycogen"  reversible="true">
       <listOfReactants>
              <speciesReference  species=" cpd00155_c "
stoichiometry="1.000000"/>
       </listOfReactants>
       <listOfProducts>
              <speciesReference  species=" cpd00155_e"
stoichiometry="1.000000"/>
       </listOfProducts>
       <kineticLaw>
              <math xmlns="http://www.w3.org/1998/Math/MathML">
                           <ci> FLUX_VALUE </ci>
              </math>
              <listOfParameters>
                     <parameter id="LOWER_BOUND" value="-10000"
units="mmol_per_gDW_per_hr"/>
                     <parameter id="UPPER_BOUND" value="10000"
units="mmol_per_gDW_per_hr"/>
                     <parameter id="OBJECTIVE_COEFFICIENT" value="0.0"/>
                     <parameter id="FLUX_VALUE" value="0.000000"
units="mmol_per_gDW_per_hr"/>
              </listOfParameters>
       </kineticLaw>
</reaction>
```

2. Set this newly added reaction as the model objective function:

   ```
   model = changeObjective(model, 'EX_ cpd00155 (e)')
   ```

3. Optimize the model for this objective function:

   ```
   FBAsolution = optimizeCbModel(model,'max')
   ```

4. If `FBAsolution.f` is greater than 0, it means that the compound under analysis can be synthesized and you can move to the next one. In case the flux value (`f`) across this reaction is 0, then one (or more) metabolic gap is present along the biosynthetic pathway leading to the production of that specific biomass precursor. In order to trace them back, repeat **steps 1–3** for each of the metabolic reactions that are involved in the biosynthesis of the biomass component (and its precursors) that cannot be synthesized until you find the missing reaction(s). Once identified, you can use comparative genomics and organism-specific databases (*see* Subheading 3.1) to fill the gap.

Once you have repeated these steps for all the biomass constituents and gap-filled the model, the model should be able to produce biomass on the growth medium defined by the LBs of exchange reactions.

3.11 Validate Model Against Experimental Data

Besides being able to produce biomass, metabolic reconstructions are also required to fit as much as possible with experimental data. To check their reliability in predicting growth phenotypes, metabolic models can be compared against large-scale growth tests (e.g., Biolog Phenotype Microarray, *see* Chapter 7) or experimentally calculated growth rates. Comparing in silico-predicted growth against data from high-throughput phenomics gives indication on the overall capability of the model to correctly predict growth on a large set of known carbon sources. As already done for other reconstructions [44–46], Biolog data and model optimization outcomes can be easily compared. To do this:

1. Collect Biolog information on known carbon sources in a spreadsheet. This should include (a) compound names, (b) KEGG compound codes, and (c) growth/non-growth phenotypes. A simple example of a valid reference file for this analysis is shown in Table 2.

Table 2
Schematic tabular representation of processed Biolog results used for comparing in vivo data with model predictions

KEGG code	Substrate	Growth
C00025	L-Glutamic acid	Yes
C00026	a-Keto-glutaric acid	No
C00031	a-D-Glucose	No
C00033	Acetic acid	Yes

2. For each of the compounds listed in the Biolog-derived table
   (a) If the compound is not present in the model, add it. Use KEGG reference code to univocally identify shared compounds between Biolog dataset and metabolic reconstruction.
   (b) Add an exchange reaction accounting for its utilization by the model (*see* Subheading 3.4).
   (c) Use LB of this reaction to set its uptake rate (if not known then use an arbitrary value, e.g., 10 mmol/g h).
   (d) Ensure that all the other LBs of exchange reactions of carbon source compounds are set to 0.
   (e) Optimize the model for biomass production and record `f` value (`f` > 0 or `f` = 0).
   (f) Check if Biolog data and model prediction agree (both growth or both non-growth phenotypes) or if they do not. In this latter scenario, two alternatives are possible, i.e., (1) Biolog records growth whereas model predicts no growth or, conversely, (2) Biolog does not record growth whereas model predicts growth. To fix point (1):
      - Check if a transport reaction for the carbon source under analysis is present within the model. If this is not the case, check the genome for a gene putatively encoding a transporter able to import the carbon source under analysis. TCDB (Transporter Classification Data Base, http://www.tcdb.org/, [47]) can be a valuable resource in this sense. An artificial transport reaction (i.e., without any associated coding genes) can be added at this point for debugging purposes.
      - If biomass is not produced (`f` = 0) even after adding the transport reaction to the model, search for possible metabolic gaps in the model by repeating **steps 1–4** of Subheading 3.10 (setting biomass production as the objective function).
      - Include a further column to the table shown above, including model growth prediction for each compound and highlight possible incongruences (*see* Table 3 for an example).

Fixing case (2) is trickier and it may involve the removal of one (or more) reaction(s) erroneously added to the model during the reconstruction process (with the risk to remove reactions that are crucial under other growth conditions) and/or an accurate revision the substrate specificity of the transporters in model.

As a rule of thumb, 80/90 % agreement between Biolog data and model predictions can be considered satisfactory; this is usually found for most of the reconstructions that have been validated against high-throughput Phenomics to date [44, 45].

Table 3
Comparison between model prediction and in vivo (Biolog) data

KEGG code	Substrate	Growth	Model
C00025	L-Glutamic acid	Yes	Yes
C00026	a-Keto-glutaric acid	No	Yes
C00031	a-D-Glucose	No	No
C00033	Acetic acid	Yes	Yes

Model-predicted growth rate (`f` value) can also be compared against experimentally determined growth rates (both expressed as h^{-1}). Specific solutions for fixing erroneous in silico predictions (either too fast or too slow predicted growth) can be found in [22].

Although the scope and the purpose of the reconstruction define whether the iterative reconstruction process can be considered "finished" [22], the model capability of synthesizing all the components of the biomass and an overall agreement between model prediction and experimental data are usually considered a first achievement in the overall reconstruction process and a reliable base for further analyses. However, since the reconstruction will not likely embed information on more than 20–30 % of the encoded enzymes, continuous effort is necessary to periodically update and revise the metabolic model and to include as much information as possible (e.g., gene-protein relationships, organism-specific reactions, experimental data).

3.12 Dynamic Flux Balance Analysis (dFBA)

Among all the possible modeling strategies, dFBA has been gaining increasing interest. Basically, dFBA combines extracellular dynamics with intracellular pseudosteady states and thus may be suitable for the simulation of metabolic behavior under dynamic conditions. dFBA provides a framework for analyzing the transience of metabolism due to metabolic reprogramming and for obtaining insights for the design of metabolic networks [48]. This technique has been widely adopted for predicting different cellular metabolic states, including the diauxic shift of *E. coli* growth [48] and the effect of genetic manipulations on ethanol production [49].

dFBA is basically an iteration of the FBA method where at each (user-defined, see below) time step FBA is used to compute the cellular growth rate together with nutrient utilization rate and (eventual) by-product efflux. These outputs are then used, at the following time point, to recompute biomass production, nutrient uptake, and by-product secretion. This iterative procedure continues until the last time point is reached. Resulting biomass, nutrients, and by-products can then be plotted in a graph accounting for the values of each of these quantities at the different time points.

dFBA is implemented in COBRA toolbox and to run it on your reconstructed model in the COBRA toolbox use

```
dynamicFBA(model,SubstrateUptake,InitialConcentration,InitialBiomass, TimeStep, NSteps,RxnsToPlot);
```

where

`model` is the metabolic model as imported into COBRA.

`SubstrateUptake` embeds the list of the reactions accounting for the uptake of the nutrients.

`InitialConcentration` is the concentration of the nutrient source at the beginning of the dFBA run.

`InitialBiomass` is the initial amount of cellular biomass.

`TimeStep` defines the size of each time step during the iteration.

`NSteps` defines how many steps will be performed during the iteration.

`RxnsToPlot` defines the list of reactions whose values will be used for plotting the results of the dFBA simulation.

4 Notes

1. Compound names of iAbaylyiV4 biomass assembly reaction are

```
<species id="cpd00001_c" name="H2O_H2O" compartment="c" charge="0" boundaryCondition="false"/>
<species id="cpd00002_c" name="ATP_C10H13N5O13P3" compartment="c" charge="-3" boundaryCondition="false"/>
<species id="cpd11461_c" name="DNA_C15H23O13P2R3" compartment="c" charge="-2" boundaryCondition="false"/>
<species id="cpd11462_c" name="mRNA_" compartment="c" charge="10000000" boundaryCondition="false"/>
<species id="cpd11463_c" name="Protein_C4H5N2O3R2" compartment="c" charge="-1" boundaryCondition="false"/>
<species id="cpd11469_c" name="(2E)-Dodecenoyl-[acp]_C23H41N2O8PRS" compartment="c" charge="-1" boundaryCondition="false"/>
<species id="cpd11677_c" name="Triglyceride_C6H5O6R3" compartment="c"
charge="0" boundaryCondition="false"/>
<species id="cpd16601_c" name="generic_fatty acid chain for free molecules (mass)_"
```

```
compartment="c" charge="10000000" boundary
Condition="false"/>
<species id="cpd16653_c" name="generic_cofac-
tor molecule (mass)_" compartment="c" charge=
"10000000" boundaryCondition="false"/>
<species id="cpd16661_c" name="generic_pep-
tidoglycan (mass)_" compartment="c" charge=
"10000000" boundaryCondition="false"/>
<species id="cpd16662_c" name="generic_phos-
pholipid (mass)_" compartment="c" charge=
"10000000" boundaryCondition="false"/>
<species id="cpd16663_c" name="generic_free
polysaccharide (mass)_" compartment="c" charge=
"10000000" boundaryCondition="false"/>
<species id="cpd16669_c" name="generic_wax
esters (mass)_" compartment="c" charge=
"10000000" boundaryCondition="false"/>
<species id="cpd00008_c" name="ADP_
C10H13N5O10P2" compartment="c" charge="-2"
boundaryCondition="false"/>
<species id="cpd00009_c" name="Phosphate_
HO4P" compartment="c" charge="-2" boundary
Condition="false"/>
<species id="cpd11416_c" name="Biomass_"
compartment="c" charge="0" boundaryCondition=
"false"/>
```

2. The enzymatic capacity (EC) for carbon source utilization is determined as the ratio of the growth rate (μ) to the biomass yield in batch experiments (biomass yield) [7]:

$$\mathrm{EC} = \frac{\mu}{\text{Biomass yield}}$$

Biomass yield is defined as the ratio of the amount of biomass produced to the amount of substrate consumed:

$$\text{Biomass yield} = \frac{g \text{ of biomass}}{g \text{ of substrate utilized}}$$

Note that, since in FBA all reaction fluxes are expressed as mmol/g h, biomass yield should be converted taking into consideration mmol of substrate provided before calculating EC with

$$\text{mmol of substrate} = \frac{g \text{ of substrate}}{\text{MW of substrate}}$$

where MW is the molecular weight of the substrate.

References

1. Downs DM (2003) Genomics and bacterial metabolism. Curr Issues Mol Biol 5(1):17–25
2. Beloqui A, de Maria PD, Golyshin PN, Ferrer M (2008) Recent trends in industrial microbiology. Curr Opin Microbiol 11(3):240–248
3. Zou W et al (2012) Reconstruction and analysis of a genome-scale metabolic model of the vitamin C producing industrial strain Ketogulonicigenium vulgare WSH-001. J Biotechnol 161(1):42–48
4. Garcia-Ochoa F, Santos VE, Casas JA, Gomez E (2000) Xanthan gum: production, recovery, and properties. Biotechnol Adv 18(7):549–579
5. George HA, Johnson JL, Moore WE, Holdeman LV, Chen JS (1983) Acetone, isopropanol, and butanol production by Clostridium beijerinckii (syn. Clostridium butylicum) and Clostridium aurantibutyricum. Appl Environ Microbiol 45(3):1160–1163
6. Lee SJ et al (2005) Metabolic engineering of Escherichia coli for enhanced production of succinic acid, based on genome comparison and in silico gene knockout simulation. (Translated from eng). Appl Environ Microbiol 71(12):7880–7887
7. Varma A, Palsson BO (1994) Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type Escherichia coli W3110. Appl Environ Microbiol 60(10):3724–3731
8. Oberhardt MA, Palsson BO, Papin JA (2009) Applications of genome-scale metabolic reconstructions. Mol Syst Biol 5:320 (in eng)
9. Orth JD, Thiele I, Palsson BO (2010) What is flux balance analysis? Nat Biotechnol 28(3): 245–248
10. Oberhardt MA, Chavali AK, Papin JA (2009) Flux balance analysis: interrogating genome-scale metabolic networks. Methods Mol Biol 500:61–80
11. Fang X, Wallqvist A, Reifman J (2012) Modeling phenotypic metabolic adaptations of Mycobacterium tuberculosis H37Rv under hypoxia. PLoS Comput Biol 8(9):e1002688
12. Park JM, Kim TY, Lee SY (2009) Constraints-based genome-scale metabolic simulation for systems metabolic engineering. Biotechnol Adv 27(6):979–988 (in eng)
13. Maarleveld TR, Khandelwal RA, Olivier BG, Teusink B, Bruggeman FJ (2013) Basic concepts and principles of stoichiometric modeling of metabolic networks. Biotechnol J 8(9): 997–1008
14. Moura M, Broadbelt L, Tyo K (2013) Computational tools for guided discovery and engineering of metabolic pathways. Methods Mol Biol 985:123–147
15. Copeland WB et al (2012) Computational tools for metabolic engineering. Metab Eng 14(3):270–280
16. Durot M, Bourguignon PY, Schachter V (2009) Genome-scale models of bacterial metabolism: reconstruction and applications. FEMS Microbiol Rev 33(1):164–190 (in eng)
17. Aziz RK et al (2008) The RAST Server: rapid annotations using subsystems technology. (Translated from eng). BMC Genomics 9:75
18. Schneider J et al (2010) CARMEN - comparative analysis and in silico reconstruction of organism-specific MEtabolic networks. Genet Mol Res 9(3):1660–1672
19. Feng X, Xu Y, Chen Y, Tang YJ (2012) MicrobesFlux: a web platform for drafting metabolic models from the KEGG database. BMC Syst Biol 6:94
20. Karp PD et al (2010) Pathway Tools version 13.0: integrated software for pathway/genome informatics and systems biology. Brief Bioinform 11(1):40–79
21. Reyes R et al (2012) Automation on the generation of genome-scale metabolic models. (Translated from eng). J Comput Biol 19(12): 1295–1306
22. Thiele I, Palsson BO (2010) A protocol for generating a high-quality genome-scale metabolic reconstruction. Nat Protoc 5(1):93–121
23. Thiele I, Palsson BO (2010) Reconstruction annotation jamborees: a community approach to systems biology. Mol Syst Biol 6:361
24. Dandekar T, Fieselmann A, Majeed S, Ahmed Z (2012) Software applications toward quantitative metabolic flux analysis and modeling. Brief Bioinform 15:91
25. Henry CS et al (2010) High-throughput generation, optimization and analysis of genome-scale metabolic models. (Translated from eng). Nat Biotechnol 28(9):977–982
26. Kanehisa M (2002) The KEGG database. Novartis Found Symp 247:91–101, discussion 101–103, 119–128, 244–152
27. Karp PD, Paley S (1996) Integrated access to metabolic and genomic data. J Comput Biol 3(1):191–212 (in eng)
28. Karp PD, Paley S, Romero P (2002) The pathway tools software. Bioinformatics 18(Suppl 1): S225–S232
29. Caspi R et al (2012) The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases.

Nucleic Acids Res 40(Database issue): D742–D753
30. Karp PD, Riley M, Paley SM, Pellegrini-Toole A (2002) The MetaCyc database. Nucleic Acids Res 30(1):59–61
31. Altman T, Travers M, Kothari A, Caspi R, Karp PD (2013) A systematic comparison of the MetaCyc and KEGG pathway databases. BMC Bioinform 14(1):112 (in Eng)
32. Hyland C, Pinney JW, McConkey GA, Westhead DR (2006) metaSHARK: a WWW platform for interactive exploration of metabolic networks. Nucleic Acids Res 34(Web Server issue):W725–W728
33. Pinney JW, Shirley MW, McConkey GA, Westhead DR (2005) metaSHARK: software for automated metabolic network prediction from DNA sequence and its application to the genomes of Plasmodium falciparum and Eimeria tenella. Nucleic Acids Res 33(4): 1399–1409
34. Liao YC, Tsai MH, Chen FC, Hsiung CA (2012) GEMSiRV: a software platform for GEnome-scale metabolic model simulation, reconstruction and visualization. Bioinformatics 28(13):1752–1758
35. Agren R et al (2013) The RAVEN toolbox and its use for generating a genome-scale metabolic model for Penicillium chrysogenum. PLoS Comput Biol 9(3):e1002980
36. Swainston N, Smallbone K, Mendes P, Kell D, Paton N (2011) The SuBliMinaL Toolbox: automating steps in the reconstruction of metabolic networks. J Integr Bioinform 8(2):186
37. Hucka M et al (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19(4):524–531
38. Bray T, Paoli J, Sperberg-McQueen CM (1998) Extensible markup language (XML) 1.0. Available from: http://www.w3.org/TR/1998/REC-xml-19980210
39. Feist AM et al (2007) A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. Mol Syst Biol 3:121
40. Schellenberger J et al (2011) Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0. Nat Protoc 6(9):1290–1307
41. Ebrahim A, Lerman JA, Palsson BO, Hyduke DR (2013) COBRApy: COnstraints-based reconstruction and analysis for python. BMC Syst Biol 7:74
42. Bornstein BJ, Keating SM, Jouraku A, Hucka M (2008) LibSBML: an API library for SBML. Bioinformatics 24(6):880–881
43. Karp PD et al (2000) The EcoCyc and MetaCyc databases. Nucleic Acids Res 28(1):56–59
44. Oberhardt MA, Puchalka J, Fryer KE, Martins dos Santos VA, Papin JA (2008) Genome-scale metabolic network analysis of the opportunistic pathogen Pseudomonas aeruginosa PAO1. J Bacteriol 190(8):2790–2803
45. Fang K et al (2011) Exploring the metabolic network of the epidemic pathogen Burkholderia cenocepacia J2315 via genome-scale reconstruction. BMC Syst Biol 5:83
46. Oberhardt MA, Puchalka J, Martins dos Santos VA, Papin JA (2011) Reconciliation of genome-scale metabolic reconstructions for comparative systems analysis. PLoS Comput Biol 7(3): e1001116
47. Saier MH Jr, Tran CV, Barabote RD (2006) TCDB: the Transporter Classification Database for membrane transport protein analyses and information. Nucleic Acids Res 34(Database issue):D181–D186
48. Mahadevan R, Edwards JS, Doyle FJ 3rd (2002) Dynamic flux balance analysis of diauxic growth in *Escherichia coli*. Biophys J 83(3): 1331–1340 (in eng)
49. Lisha KP, Sarkar D (2014) Dynamic flux balance analysis of batch fermentation: effect of genetic manipulations on ethanol production. Bioprocess Biosyst Eng 37(4):617–627
50. Overbeek R et al (2005) The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes. Nucleic Acids Res 33(17):5691–5702
51. Boele J, Olivier BG, Teusink B (2012) FAME, the flux analysis and modeling environment. BMC Syst Biol 6:8