Modeling and Simulation Tools: From Systems Biology to Systems Medicine

Systems Medicine pp 441-463 | Cite as

- Brett G. Olivier (1) Email author (b.g.olivier@vu.nl)
- Maciej J. Swat (2)
- Martijn J. Moné (3) (4)

1. Systems Bioinformatics, VU University Amsterdam, , Amsterdam, The Netherlands
2. EMBL-European Bioinformatics Institute, Wellcome Trust Genome Campus, , Hinxton, UK
3. Molecular Cell Physiology, VU University Amsterdam, , Amsterdam, The Netherlands
4. Systems and Synthetic Biology, Wageningen University, , Wageningen, The Netherlands

Part of the Methods in Molecular Biology book series (MIMB, volume 1386)

Abstract

Modeling is an integral component of modern biology. In this chapter we look into the role of the model, as it pertains to Systems Medicine, and the software that is required to instantiate and run it. We do this by comparing the development, implementation, and characteristics of tools that have been developed to work with two divergent methodologies: Systems Biology and Pharmacometrics. From the Systems Biology perspective we consider the concept of “Software as a Medical Device” and what this may imply for the migration of research-oriented, simulation software into the domain of human health.

In our second perspective, we see how in practice hundreds of computational tools already accompany drug discovery and development at every stage of the process. Standardized exchange formats are required to streamline the model exchange between tools, which would minimize translation errors and reduce the required time. With the emergence, almost 15 years ago, of the SBML standard, a large part of the domain of interest is already covered and models can be shared and passed from software to software without recoding them. Until recently the last stage of the process, the

pharmacometric analysis used in clinical studies carried out on subject populations, lacked such an exchange medium. We describe a new emerging exchange format in Pharmacometrics which covers the non-linear mixed effects models, the standard statistical model type used in this area. By interfacing these two formats the entire domain can be covered by complementary standards and subsequently the according tools.

Key words

Systems biology Software design Standards development SBML Kinetic modeling Constraint-based modeling Quantitative and systems pharmacology Physiology-based pharmacokinetics Pharmacodynamics Pharmacometrics

1 Introduction

For more than a decade the Systems Biology approach has led to an integration of theoretical, modeling, and experimental approaches directed toward the understanding of complex biological systems. In this process, modeling has become a key component of Systems Biology and is integral to both quantitative “bottom-up” and qualitative “top-down” approaches [1]. While the former includes detailed mechanistic approaches such as kinetic [2] and constraint-based modeling [3], the latter includes qualitative methods that include Boolean and petri-nets modeling [4] and statistical inference [5]. However, supporting this modeling process is the underlying assumption that there exists software in which a mathematical model can be instantiated and interrogated by a user, thereby generating results ready for further analysis. This second process involves the development of simulation software, the implementation of new theory, and the use of standards for data model description and exchange.

While the Systems Biology community has an established record of tool development, in general, much of this software has been developed in academia as a tool designed primarily to answer a specific research question or as a vehicle to illustrate a newly developed analysis method or algorithm. In contrast, general-purpose simulation software provides an integrated package of modeling and simulation methods that can, generally, be applied to a specific class of model or modeling methodology. This often leads, unsurprisingly, to the situation where the insight gained from applying the software to a research problem is considered more important than the software development process itself.

In this chapter we will first look at the definition of “a model” as it applies to Systems Biology (SB), Quantitative and Systems Pharmacology (QSP) , and finally to Pharmacometrics (PMX). We examine the various strategies used to encode them, software used to run them and investigate how these are relevant to the development of tools for Systems Medicine. We will discuss key issues, such as user interface, model

description and instantiation, software architecture, and standards support, that should be considered when selecting or designing a modeling tool. An aspect that is specifically relevant for Pharmacometrics and which will be discussed in detail is that of an extended model definition as a result of the use of population datasets required in clinical context. Starting from two divergent perspectives we attempt to address a common question: "What is required of current and future software (SB, QSP, PMX) such that it is relevant for Systems Medicine?"

2 Models and Systems Medicine

Systems Medicine encompasses the "iterative and reciprocal feedback between clinical investigations and practice with computational, statistical and mathematical multi-scale analysis and modeling of pathogenic mechanisms, disease progression and remission, disease spread and cure, treatment responses and adverse events as well as disease prevention both at the epidemiological and individual patient level" [6]. As a result, going toward Systems Medicine means medicine will move away from reductionist concepts, and toward holistic understanding of health and disease. The successful and assessable outcome would be a medical practice that revolves around systems-based approaches and that becomes more and more predictive. Therefore, Systems Medicine aims to implement Systems Biology approaches in medical concepts and research, as well as in medical practice. It is important to underline that in Systems Biology, models are key in explaining and predicting biological phenomena, and are as such indispensable for the process of resolving biological problems or research questions. The dynamic data integration that is required to account for the biological complexity cannot be done without computer-assisted analyses and simulations. As a consequence, the transition of Systems Biology to Systems Medicine in practice will, from the start, revolve around the use of models.

In biology, the term "model" can be used for any description that proposes to explain or represent a part of biological reality. This can then be anything, for example a cartoon depicting a suggested binding mechanism of some molecule to an enzyme. Although such models and other basic representations will remain useful in both biology and medicine, in Systems Biology and Systems Medicine, and hence in this chapter, when we refer to models we explicitly assume them to be mathematical, computable representations that reflect (a part of) biological reality.

In addition, there is a wide variety of different types of mathematical models and modeling approaches in Systems Biology, and many of these can be valuable for application in a medical context. Some of these are mechanistic, e.g. kinetic molecular models that integrate biochemical and biophysical processes, while others may be non-mechanistic, like for instance models based on statistical and machine learning theory for the analysis of large-scale data. In addition, models may operate at different scales, either in time or in space. Previous chapters have addressed case studies that illustrate where and how several different approaches can be applied in manners that are suitable for Systems Medicine. It is not within the remit of this chapter to provide an exhaustive overview of which modeling techniques are available in Systems Biology and whether or not they could be valuable for Systems Medicine. From the notion that mathematical

models are an essential part for the successful implementation of Systems Medicine, follows that the software tools that are needed to instantiate and operate the models will constitute an indispensable part in Systems Medicine. Therefore, we focus on a subset of existing Systems Biology modeling software tools, which are used in mechanistic molecular Systems Biology, and make clear that they can well serve as a proxy for considerations on Systems Medicine requirements of mathematical modeling software tools in general.

Modeling, i.e. the verb as it is used in biological and biomedical research fields, is commonly understood to be the activity of constructing some model by integrating all the relevant knowledge and data and running simulations on them, i.e. computing (desired) outcomes from the mathematically encoded model given a set of starting and/or boundary conditions. In order to consider how these activities essentially relate to the use of the underpinning software, and what the repercussions are for such software to be applicable in a medical setting, it is illuminating to break down the process from model encoding via model instantiation to dealing with the modeling outputs. This brings to light that considering modeling tools for medical use is not merely relying on the computational strengths and weaknesses of the tool per se, but moreover about requirements with respect to the software and tool structure itself.

As can be appreciated from previous chapters, the clinical and medical sectors are broad and heterogeneous. The requirements for modeling and modeling software will therefore not be unambiguous and depends to a large extent on the precise context within which it will be used. In pursuance of providing concrete insights into issues concerning the use of modeling software for Systems Medicine, it is useful to distinguish three major environments within the field. First, the academic research environment; this is where most of the Systems Biology is routinely performed, and where most of the current modeling tools are developed. Its research outcomes can have clinical and medical relevance, but mostly at the conceptual level as it does not have to deal with real-world clinical and medical issues. Next, there is the medical and clinical research environment, which focuses, for example, on drug target discovery and novel therapeutics, and has to deal with controlled clinical trials. And third there is daily medical practice, which includes for instance patient diagnosis, treatment decisions, and ultimately Personalized Medicine. In viewpoint discussions on Systems Medicine there has been much emphasis on Personalized Medicine, or P4 medicine [7]. Nonetheless, before personalizing for example drug treatment, the development of new drugs remains a prerequisite. There is a clear requirement for new drugs in treating, e.g. cancer, Alzheimer's, osteoporosis, to name but a few, but we also need novel antibiotic and antiviral drugs. It is here, in the medical and clinical research environment, where most of the modeling-driven research approaches are being put into clinical practice. On the one hand to aid the drug development pipeline, but also to predict population level drug effects using Pharmacometrics. Modeling concepts and approaches from Systems Biology are actively being carried out here, so in the last part of this chapter we will focus in more detail on the current state of modeling deployment in this field with a focus on the supporting software that is being utilized.

3 Selecting a Systems Biology Modeling Tool

In the previous section, we have seen that models and modeling are as central to Systems Medicine as they are to Systems Biology. As stated above, in this section we assume that "a model" is a quantitative, mechanistic description of a biological system that can be described as a set of coupled ordinary differential equations (ODEs). Furthermore, "a model" is assumed to be either a detailed kinetic description of such a system, usable for both time simulation and steady-state analysis [8, 9, 10, 11], or a genome scale constraint-based model suitable for use in flux balance analysis (FBA) [12, 13]. Both these modeling formalisms are widely used in Systems Biology and supported by large, and active, software and standards development communities. In this section we consider the modeling process and look at the different strategies that can be used to design or select a modeling tool such that a model can be encoded, simulated in software and exchanged between tools. We then consider a hypothetical use case where a model is used as a diagnostic tool in Systems Medicine, a role that could be considered a medical device. Finally, we compare a selection of Systems Biology tools that highlight a variety of the aforementioned strategies. Figure 1 illustrates how software and standards can be combined to enable the transition from Systems Biology to Systems Medicine. To begin with let us consider the question: "How does one encode a model such that it can run on a computer?"

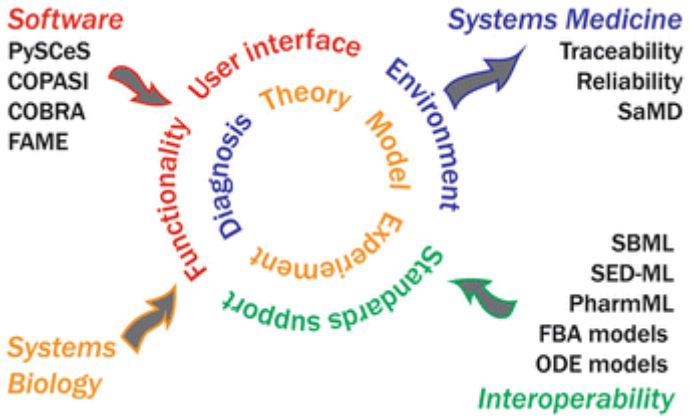

Fig. 1

Overview. The development of systems medicine will require new levels of software interoperability, expanded use of standards, relevant tool design, and a novel application of systems biology. Abbreviations used in this diagram are defined in the main text

3.1 Model Encoding

A widely used method for encoding a quantitative model is to write it as a human readable set of mathematical equations (e.g. as ODEs). If the model is to be simulated on a computer then this mathematical description still needs to be understood by a modeler and encoded in a programming language that can be interpreted by a computer. Though widely used, this approach has many points of failure, for example, the accuracy of the model writer, the knowledge and skill of the person interpreting the mathematics and those responsible for writing the code that is then compiled and executed. A slightly higher-level approach is to encode the model directly in a computer language such as

Fortran, C, C++, Java, or Python. While this has the advantage that, in principle, no human intervention is required to translate the model into a machine usable format, it makes understanding the model content and its validation (and debugging in the case of errors) very difficult for anybody not involved with the initial encoding effort. Another problem with this approach is that the model description starts to utilize features, or take on the characteristics, specific to the particular language that it is encoded in, e.g. usage of data structures and functionality provided by the programming language itself. In this way there is the potential that the model quickly becomes non-portable, i.e. not easily translatable into a format usable in any other software. A final consideration is that neither of these approaches scale well to the encoding of larger, more complex, problems, nor do they intrinsically support a portable method of annotating the individual model components.

Interactive model encoding, or the use of a custom model description language, are two approaches to model encoding often employed by standalone simulation software. Tools utilizing only a graphical user interface (GUI) generally allow users to create or edit models directly within the interface itself. A good example of this GUI-based approach is COPASI [14], while visual design tools such as JDesigner, described later in this section, offer an even higher-level interface. Alternatively, tools employing command line interfaces (CLIs) often make use of a text-based, human and machine readable, model description language. Examples of simulation tools that use this approach are JARNAC [15] and PySCeS [16], both of which use a format originally developed for use by SCAMP [17]. An analogous approach, often used in constraint-based modeling tools, e.g. CellNetAnalyser [18], is to describe the model as separate ASCII files containing lists of model components. While using a GUI is arguably the user-friendliest way of inputting a moderate-sized model typical of the average detailed kinetic model, it does not scale to the creation of large-scale models. In this case a text-based model description language may be more appropriate with the associated disadvantage that such approach can be error prone, especially for non-expert users. For very large-scale models such as genome scale reconstructions (GSRs) , text or table-based model descriptions become necessary to deal with the encoding of hundreds or thousands of model components [19].

While the model encoding methods described above cover a range of use-cases, almost all of them are specific to a single tool or family of tools. In today's Systems Biology software landscape, there are many different tools offering a wide range of functionality, the inability to exchange model descriptions between tools is a singular disadvantage. In a later section we will discuss interoperable model exchange format; for now we address the question "how do we instantiate a model in software?"

3.2 Model Instantiation

At this stage "the model" is a document or structured dataset that can be found in an online database such as BioModels [20, 21]. At minimum the model can be interpreted as a structural representation of a biological network analogous to a DNA sequence stored in GenBank [22]. If the model components are sufficiently well annotated, it can be used as a structured dataset suitable for use in text mining and semantic analysis [23]. However, to fully realize a quantitative model's function it needs to be interpreted and

instantiated in software on a computer. A variety of strategies can be employed for this purpose:

- Both model and analysis methods are directly implemented in a general programming language such as FORTRAN, C, C++, Java, or Python [24].
    - Analysis methods are implemented in a mathematical environment, e.g. MATLAB™, R, Mathematica™, or SciPy (scipy.org), using the provided built-in tools.
    - An existing mathematical environment is extended with additional functionality, for example, MathSBML for Mathematica™ [25] or the COBRA Toolbox for MATLAB™ [26].
    - The model is instantiated in dedicated simulation software, developed as either standalone software like COPASI or as a web-based application like JWS Online [27].

All of these strategies have their own advantages and disadvantages. Directly encoding a model in a programming language or mathematical environment gives one complete control over its analysis, yet this process can be extremely time-consuming, can be error prone, and requires specialist knowledge of a programming language and numerical analysis. On the other hand, using dedicated simulation software may provide the required functionality at the "click-of-a-button," yet not provide the type of analysis required by the user and be difficult to extend with new functionality. Commercial mathematical environments such as MATLAB™ or Mathematica™ may provide support for their supplied algorithms but are not freely available outside of certain business sectors or academia (through academic licensing programs).

This interplay between high-performance commercial products and free open software also takes place at the algorithm implementation level. Consider a mixed-integer linear program (MILP) solver that is becoming integral to software performing constraint-based modeling and analysis. While there are free solvers available, for example GLPK (www.gnu.org (http://www.gnu.org/)), for very large or complex models or models that require advanced analysis a commercial solver such as IBM CPLEX™ [28] or Gurobi™ [29] becomes necessary or even required. As licenses for these commercial solvers can be expensive, they are available in specialist commercial enterprises and academia (again through free licensing programs).

It is therefore of utmost importance to be aware that the strategy chosen to run a model in silico is both highly dependent on what analysis will be performed on the model, as well as the end-user environment where the software will be deployed. Once an appropriate strategy is chosen it can be used to assist in the selection of an existing simulation tool or in the design of a new one.

We previously highlighted that, in principle, every tool has its own encoding format, or way of inputting a model. Using one of the above strategies, “the model” is then instantiated in the software and can be used and interrogated. However, what if we want to exchange a model between different software tools?

3.3 Model Exchange

The ability to exchange models has become an important feature of Systems Biology software that has independently developed across a diverse community of researchers [30, 31]. As quantitative models have increased in size and complexity and their use has become more widespread in the life sciences it has become critical to use them in ways not necessarily thought of by their original authors. For example, components or processes could be used in a different context or recombined with other independently developed models. In order to facilitate these processes it is vital that any model component’s identity can be unambiguously established and that it can be annotated with context-specific information.

In addition the number of analyses that can be performed using a model is rapidly expanding and no single modeling tool can incorporate all of them. Instead, multiple, individually developed, highly specialized tools will be required to work together to perform the “next generation” of simulation experiments. To do this will require the ability to seamlessly exchange models between different software, or, in other words, complete tool interoperability—independent of tool development language or operating environment. Standardized data formats can provide a platform that facilitates both of these processes, especially, when the foundations of this emerging tool interoperability are the development of open, community-driven standards and their implementation in both research and industrial software.

In the last 15 years or so, a set of standards for encoding and working with a variety of modeling methodologies have evolved in the Systems Biology community. While most of these standards have developed independent of one another, recent initiatives such as the *COmputational Modeling in BIology NEtwork* (COMBINE) aim to enhance the interaction among existing standards and facilitate the development of new ones [32]. COMBINE (co.mbine.org) incorporates a diverse range of standards and associated standardization efforts that include BIOPAX, a standard for the description of biological pathways, CellML for biological and physiological models [33] and the Synthetic Biology Open Language (SBOL) for exchanging genetic designs [34]. Many of the COMBINE standards incorporate aspects of two minimal guidelines relevant to a model or simulation. The first of these, the *Minimal Information Required In the Annotation of Models* (MIRIAM), is a set of guidelines that has been developed for the consistent and persistent annotation and curation of computational models in biology. MIRIAM can be implemented in any structured model format where individual components can be annotated [35]. The Minimal Information required for a Simulation Experiment (MIASE) provide the modeler or developer with a set of guidelines on the information necessary to reproduce a simulation experiment [36]. In the next section, we will look in more detail at two COMBINE standards that together implement both the MIRIAM and

MIASE guidelines and are particularly relevant when selecting a kinetic or constraint-based modeling tool. The web addresses for some of the tools mentioned in this section are provided in Table 1.

Table 1

Relevant URLs referred to in the text

SBML Test Suite	sbml.org/Software/SBML_Test_Suite
SBML Validator	sbml.org/Facilities/Validator
SBML Software matrix	sbml.org/SBML_Software_Guide
libSEDML	libsedml.sourceforge.net/libSedML/Welcome.html
JlibSEDML	sourceforge.net/projects/jlibsedml
SED-ML Web Tools	sysbioapps.dyndns.org/SED-ML_Web_Tools
COMBINE archives	co.mbine.org/documents/archive
ADAPT II	bmsr.usc.edu/software/adapt
Monolix	lixoft.com
NONMEM	www.iconplc.com/technology/products/nonmem
Phoenix NLME	www.certara.com/products/pkpd/phx-nlme
WinBUGS	www.mrc-bsu.cam.ac.uk/software/bugs
PharmML	ddmore.eu/pharmml and pharmml.org

The Systems Biology Markup Language (SBML) is a widely used standard for the interoperable encoding of Systems Biology models and is a machine-readable data format that can be used to encode biological processes and serialized, or written down, in the widely used eXtensible Markup Language (XML) [37]. SBML goes beyond pathway description and like CellML encodes the network reaction structure, the mathematical

equations that describe the biological processes and the numerical values of model parameters. Furthermore, SBML provides an annotation mechanism that allows the model and each of its components to be fully annotated through a MIRIAM compliant annotation mechanism taking advantage of the resource and stable identifiers provided by the MIRIAM registry and the associated online resource, "identifiers.org" [38, 39].

The SBML standard is itself evolving with major advancements referred to as "levels." For kinetic simulation software Level 2 is, generally, more widely supported, with the equivalent Level 3 Core becoming more popular. One of Level 3's advantages is that the language has been modularized so that the core model description can be extended by packages. Each package can be used alone or in combination with others to extend the core specification allowing new model types to be encoded. A good illustration of this is constraint-based models. While most existing SBML models are encoded in an SBML Level 2 dialect, the Level 3 *Flux Balance Constraints* (FBC) package provides all the necessary components required to encode a constraint-based or FBA model by extending SBML Level 3 Core with constructs describing, amongst other things, flux capacity constraints and objective functions [40]. Other packages developed include the *hierarchical composition* or combination of sub-models and the encoding of *qualitative models* such as petri-nets [41]. For tool developers, the SBML community makes free libraries available that provide bindings to multiple programming languages including C, Java, Python, MATLAB, and even Javascript. This allows software developers to easily access capabilities such as reading, writing, and modifying models using well maintained and supported libraries like libSBML [42] and JSBML [43]. In addition, SBML has a well documented and formally defined development process and management structure overseen by "editors" elected from the wider community. A comprehensive test suite that covers the interpretation of all aspects of the language, facilities for running the tests and a database of test results as well as a publicly available online validation tool are examples of the wide range of facilities provided to the modeling community.

The Simulation Experiment Design Markup Language (SED-ML) aims to implement the MIASE guidelines and is an important step along the path toward reproducible simulations. Whereas data formats such as SBML and CellML encode a model, SED-ML encodes a simulation experiment. At its core a simulation experiment consists of a quantitative model description, the set of parameter values which should be applied to the model, the algorithm that should be used to simulate the model and how the simulation output should be transformed in order to produce an expected numerical result [44]. In principle, SED-ML is independent of the format used to encode the model but in practice requires an XML-based model description. The initial SED-ML specification provided a comprehensive description of time course simulations, while recent versions have extended its capabilities to include steady-state parameter scans and now includes parameter optimization and, soon, constraint-based modeling. While there is a broad acceptance that SED-ML has an important role to play in the modeling process, up until now it has had a limited implementation even in kinetic modeling software. However, uptake should improve with the development of language libraries like libSedML and JlibSedML and community resources such as the SED-ML Web Tools (*see* Table 1).

3.4 Simulation Software as a Medical Device: A Thought Experiment

Up until now the strategies and standards discussed have been focused primarily on research-oriented tools that have primarily been developed in an academic environment. However, the question now arises: "Are these tools suitable for use, as is, in Systems Medicine?" To answer this question we need to look at the ultimate usage of the model and its software instantiation in a simulation tool. In this thought experiment we make the hypothetical assumption that within the scope of Systems Medicine there will be the possibility of using a detailed quantitative model (as previously defined) that is simulated, in software, to perform some form of diagnostic or other human health-related function.

If we now consider the following definition from *The International Medical Device Regulators Forum* (www.imdrf.org (http://www.imdrf.org/)) of Software as a Medical Device:

> The term "Software as a Medical Device" (SaMD) is defined as software intended to be used for one or more medical purposes that perform these purposes without being part of a hardware medical device [45].

Furthermore it notes (amongst other things) that:

- SaMD is a medical device and its definition includes in-vitro diagnostic medical devices.
- SaMD is capable of running on general-purpose computing hardware that has not necessarily been, specifically, designed for medical use.
- SaMD may be interfaced with other medical devices, including hardware medical devices and other SaMD, and other general-purpose software.

In our hypothetical example, the hardware could be a standard desktop computer, using a generic operating system, the input might be provided by a physician, the software could be a simulation program, the model some form of constraint-based or kinetic model, and the output some suggested diagnosis or treatment. As we are clearly using the model and its software instantiation for a medical purpose one could make the case that it easily falls within the scope of the SaMD definition.

What differentiates our new software device from existing academic simulation tools is the fact that its output can have an effect on an individual's health and thereby introduces the concept of risk (to a patient). The question of risk management in medical

devices is not new and is the subject of, amongst others, the IEC 62304 standard [46] that focuses on how the software development process should be structured and applied to manage the risk associated with developing a medical device. It does so by defining the majority of the software development and verification activities including development planning, requirement analysis, architectural design, software design, unit implementation and verification and release [47]. While an in-depth overview of IEC 62304 is beyond the scope of this chapter, one important aspect in the management of risk in the software development process is that of traceability [48]. Gotel et al. [49] define requirements traceability as:

> [...] the ability to describe and follow the life of a requirement, in both a forwards and backwards direction (i.e., from its origins through its development and specification to its subsequent deployment and use, and through all periods of on-going refinement and iteration in any of these phases).

In practice traceability can be applied to every aspect of the software development process, from formulating requirements through software documentation and version control to testing the final implementation. Therefore, traceability, risk management, and its implementation in the entire software development process will be an important factor in the potential adaptation of existing Systems Biology tools into software medical devices. Of course this is a specific and hypothetical example, but it serves to highlight that changing the role of a piece of software from research to diagnostic, may not necessarily be as straightforward as changing the input data or interpreting the output in a different way.

3.5 An Overview of Some Systems Biology Tools

Any web search for Systems Biology tools or software will return hundreds of potential matches and the user is referred to reviews of constraint-based modeling tools [50] and online resources such as the SBML software matrix for more extensive overviews of the field (Table 1, Fig. 2). The following set of tools has been selected to highlight a variety of the strategies introduced earlier with particular reference to the following features: functionality, target audience, user interface, model definition and input, support for standards, and traceability.

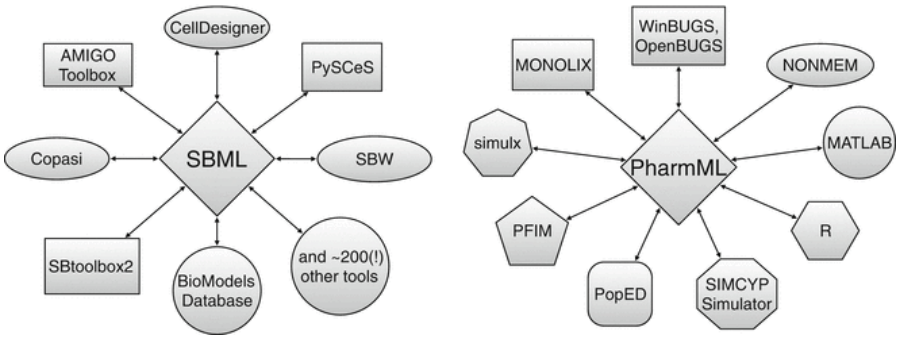

Fig. 2

Comparison of interoperability in Systems Biology (*left*) and Pharmacometrics (*right*) achieved by the exchange formats SBML and PharmML, respectively

COPASI (www.copasi.org (http://www.copasi.org/)) is an example of standalone software that has been developed for the simulation and analysis of dynamic biochemical reaction networks and aims to be usable by both beginners and advanced users through the use of a rich graphical user interface that runs on various operating systems. COPASI provides, amongst other things, stochastic and deterministic simulations as well as metabolic control analysis and parameter estimation. In addition, it also has a command line version (CopasiSE) and support for grid architectures [51]. Models can be created directly in the GUI or imported and exported in either SBML Level 1, 2, or 3, while support is also provided for MIRIAM-compliant model and component annotation. In terms of traceability, COPASI is Open Source Software that utilizes GitHub for version control. It also has extensive online documentation including version change logs, user forums, technical specifications, and video tutorials.

The *COnstraint-Based Reconstruction and Analysis Toolbox* (COBRA) is a widely used platform for constraint-based modeling [26] and the analysis of genome-scale stoichiometric models. It is aimed at an audience with intermediate to advanced technical skills, as it is a MATLAB™-based tool that can be used interactively or scripted to perform various advanced analyses. COBRA (opencobra.sourceforge.net) provides a wide range of optimization functionality that leverages either free solvers such as GLPK or commercial ones such as MOSEK™ via a MATLAB™ interface. It also includes reconstruction methods such as network gap filling and flux visualization and has recently been extended with a Python version—COBRApy [52]. COBRA can load and save models in its own native formats and model building is either scripted or interactive and supports its own version of SBML Level 2 for model import and export. Due to its wide usage, COBRA-generated SBML is one of the most widely used formats for exchanging constraint-based models. COBRA provides a platform for the addition of user-defined modules and user support is provided through online API documentation and user forums while the source code is kept under version control on SourceForge and GitHub.

The Systems Biology Workbench (SBW) [53, 54] is a framework that allows independently developed software applications to interact and exchange data using a high-performance message passing system. The basic SBW package (sbw.sourceforge.net) contains a number of tools for model creation (JDesigner), simulation (RoadRunner), and analyses (e.g. Jarnac) that allow the user to perform, amongst other things, metabolic control analysis, structural analysis, network visualization, and FBA. SBW modules generally utilize GUIs, making them well suited for use by a wide range of users. Models can be visually created (by dragging and dropping components) in JDesigner, the Jarnac model description language, or Antimony [55]. Various levels of SBML are supported for model exchange as well as MIRIAM compliant model annotation. Programmable APIs are provided for advanced users and much of the functionality is made available as web-services. SBW is developed as Open Source Software and maintains its source code on SourceForge with documentation provided as individual help files.

The *PySCeS Constraint-Based Modeling platform* (CBMPy) and *Flux Analysis and Modeling Environment* (FAME) are independent tools that together provide both a standalone and web-based modeling solution. CBMPy (cbmpy.sourceforge.net) is a Python-based cross-platform framework for constraint-based modeling that provides a range of functionality using either commercial (CPLEX™) or free (GLPK) solvers. It is targeted toward more advanced modelers and algorithm developers but also provides terminal-based GUIs for simplifying tasks such as creating reactions. CBMPy supports importing and exporting models using the latest SBML Level 3 FBC standard as well as the older COBRA SBML Level 2 format. Models can be created using Python dictionaries, Excel spreadsheets, or interactively on the command line. CBMPy supports MIRIAM annotation of model components and the export of models as COMBINE archives. Designed using a flexible architecture, CBMPy exposes its functionality as SOAP-based web services using the PySCeS-Mariner extension. CBMPy and Mariner are both Open Source Software with the source code available from SourceForge. Documentation is limited and mostly consists of API references and installation instructions.

CBMPy's flexible design allows its functionality to be utilized by the web-based modeling environment FAME [56]. FAME (f-a-m-e.org) is targeted toward both beginner and intermediate users by way of a user-friendly graphical interface. It provides facilities for creating models based on KEGG pathways [57] and editing directly in the user interface, as well as importing and exporting SBML. FAME allows the results of optimizations to be visualized on either KEGG or user-supplied pathways. It is Open Source Software and both documentation and a tutorial are available. Interestingly, by coupling CBMPy and FAME) using standard web-services, its combined functionality could be extended while still maintaining flexibility in each tool's separate development process.

This concludes the section on Systems Biology software. We now change perspective and look at how the modeling tools used in Systems Pharmacology can be relevant to Systems Medicine.

4 From Drug Discovery to Patient or from Systems Biology to Pharmacometrics

This section will focus on drug discovery and development and Pharmacometrics—the quantitative analysis of drug effects at the population level with special attention to the variability of drug responses, influences of covariates, and trial design [58]. After a short description of the typical clinical phases, we will list the corresponding areas of scientific computing associated with each of these phases [59] *see*, Fig. 3.

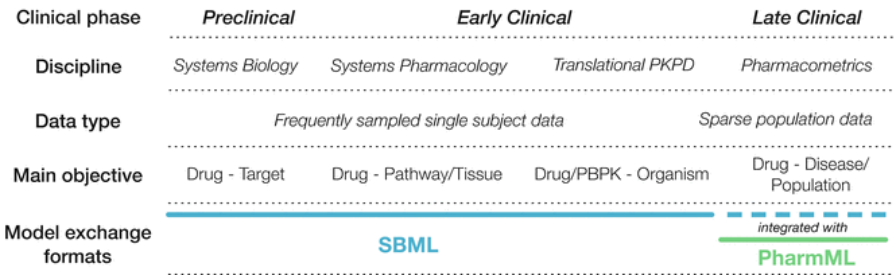

Fig. 3

Clinical phases and corresponding model exchange formats. The *dotted SBML line* indicates that it covers only the structural models. PharmML provides the missing statistical layer required for the NLME models. See text for more details

We start with the Exploratory/Discovery phase, which is when on one side the attempt is made to model basic biological processes, such as gene transcription and translation, cell cycle, signaling pathways, metabolic networks, etc., while on the other hand potential drug targets are analyzed, their influence on the whole biological process under consideration, and required receptor occupancy and response magnitudes are quantified. These phases are typically associated with Systems Biology and Quantitative and Systems Pharmacology (QSP). The latter considers the drug molecule as the major actor under investigation. It tries to validate targets and uncover mechanisms of action of existing therapeutics as well as discovering new ones [59].

Early clinical phases deal with candidate drug molecules and the whole organism, and are the bridging phase between preclinical discovery phases and large subject cohort studies of phase 3. It is here that drugs on predominantly healthy volunteers are tested, and where their bioavailability and basic PK parameters are assessed. At this stage Physiology-based Pharmacokinetics (PBPK) is the additional tool—besides QSP—that researchers have at their disposal [60, 61, 62]. The aspects of interest that PBPK can answer are summarized as ADME: absorption, distribution, metabolism, and excretion of drugs. Their understanding leads to readouts such as systemic exposure, concentration on the site of action, assessment of drug–drug interaction, etc., which are dependent on genetic constitution, disease, or sub-population ethnicity, amongst many other factors [62].

Late clinical phases look at the relationship between the disease and population by analyzing large subject cohorts. Main points of focus are safety and efficacy for the applied therapeutic dose. The approach of choice at this stage is Pharmacometrics. An essential aspect of PMX is that it is able to handle population data in contrast to SB and QSP, which require frequently sampled individual subject data. Population data in late clinical phases are often sparse, rendering SB and QSP unsuitable. PMX, on the other hand, is designed to utilize such data, constituting, in extreme cases, just one or two measurement records per subject. PMX applies statistical models that often revolve around the same deterministic prediction model as those used for SB or QSP [63, 64].

Before we turn our attention to the more technical aspects, it is worth looking at the role and effects that the application of computer models has had on the field to date. The attrition rates for new compounds vary strongly between development phases, but one can see that they also vary over the last years with a clear decreasing tendency [65]. Modeling and simulation influence is seen as the major reducing factor in PK-related attrition rates, falling between 1991 and 2001 from 40 % to as little as 10 % [66]. More modeling is required in pharmacodynamics (PD) , especially as our understanding of the biology is improving and formulating this knowledge as mechanistic models finds its way to publicly available resources like the BioModels database [20] with ready-to-use, validated, and annotated mathematical models.

4.1 Extension of the Model Definition

Definition of a model as used in SB was discussed in detail previously. Similar definitions are used in QSP and PBPK even though the complexity of the latter is often much higher, containing frequently hundreds of ODEs and algebraic equations [61]. The model there is understood to define a deterministic prediction for simulations of a time course for a variable of interest.

PMX, on the other hand, utilizes statistical models (called non-linear mixed effect (NLME) models) in the majority of cases, and requires several extensions of this model definition [64]. The continuous data model, which consists of a deterministic prediction model as used in SB and QSP, now called *Structural Model* , is enclosed by a statistical layer with a number of NLME-specific model components described below. For discrete data, very common as endpoints in clinical trials, such deterministic prediction model is optional and only the suitable distribution needs to be defined in the *Observational Model*, e.g. Poisson distribution for count data, categorical distribution for categorical data and hazard or survival function for time-to-event data [63, 64]. The next element, the *Covariate Model*, describes the relationship between PK parameters and covariates. Covariates can be either demographic (e.g. age, body weight, sex), marker of organ functions (e.g. creatinine clearance, bilirubin), environmental indicators (degree of compliance, smoking status, concomitant medication), or other factors (pregnancy, disease progression state, genotypes, and phenotypes) [63]. The *Parameter Model* allows implementing the relationship between model parameters, fixed and random effects, and covariates. The *Variability Model* describes the parameter and residual error related, unexplained variability at a certain variability level, which is often related to the structure and features of the trial design.

Pharmacometrics distinguishes another characteristic model element, the *Trial Design Model*, which is an essential component for simulation and design optimization tasks [67]. In the latter case it is used to inform the drug development process about the optimal setup of the trial with respect to the number of patients, drug doses, and other factors. It is essential in the attempt to improve the treatment efficacy and to lower its costs.

4.2 Standards and Tools

SBML has proven very valuable in SB and QSP and became a *de facto* standard for model exchange across the field in both academia and pharma [68]. Looking from the perspective of Pharmacometrics, however, it lacks support for the model elements described above. The need for a comparable exchange format has been the stimulus for Drug Disease Model Resources (DDMoRe), an on-going Innovative Medicines Initiative (IMI) project that develops an interoperability platform for tools used in PMX. One of its crucial elements is the new exchange format called Pharmacometrics Markup Language [69], of which the second public version, PharmML 0.6, has been released in January 2015 (see PharmML related websites) and containing the missing model components discussed in the previous section (Table 1). Under development is the interface for SBML-encoded models, which would provide the bridging element enabling the coverage of the entire drug discovery and development pipeline with two complementary model exchange formats. In other words, a mechanistic model developed for pre- or early-clinical use could then be taken forward and reused in the population context of late-clinical studies. This would have the advantage of seamless and error-free transition among hundreds of tools across the corresponding fields (Fig. 3). Next, we will briefly describe currently used tools for PBPK and PMX, and pointing out their most essential features (Fig. 2).

4.2.1 PBPK Tools

Simcyp Simulator has a free academic license and is centered on sophisticated ADME (absorption, distribution, metabolism, excretion) and ADAM (advanced dissolution absorption and metabolism) modules and CYP-related metabolism in liver and other organs [70]. Its main merit lies not only in the ability to rescale in vitro data on drug metabolism and transport into in vivo data, but also in the incorporation of functional genetic polymorphism information resulting in differences in, for instance, hepatic clearance in a virtual patient population of interest [61]. It contains an extensive population library, including North European Caucasians, Japanese, healthy volunteers and several other datasets for specific populations and diseases with corresponding pharmacogenetic, physiological, and biochemical data. A pediatric module and pharmacodynamics , and drug–drug interaction modules complete the tool to a feature-rich, powerful simulation, and optimization platform. A recently added custom scripting facility further extends the capabilities of the Simcyp Simulator allowing encoding of sophisticated mechanistic disease models.

Physiology-based pharmacokinetics platform **PK-Sim/MoBi** has the ability to provide highly accurate PK profiles for over thirty organs, tissue and blood compartments in combination with user-defined mechanistic PD or disease models [71]. In addition to its anatomical and physiological features, it gives valuable insights into drug metabolism processes, e.g. in liver, as related to CYP enzymes, metabolic drug–drug interactions, or the occurrence of different metabolizers. The tool also supports scenarios where, for example, a drug's active metabolite is formed in the liver by an enzyme that has a known polymorphism, which alters the metabolization rate. This opens up new possibilities for simulations starting from gene expression up to whole organs and their pathology, in a

multi-scale bottom-up approach. Similarly to the Simcyp Simulator, PK-Sim/MoBi comes with population libraries and pharmacodynamics, and pediatrics modules. Moreover, the extensibility of this platform has increased through interfacing with R and MATLAB along with an SBML interface. Comparable feature sets can be found in the commercial package **GastroPlusTM** (Simulations Plus Inc.).

4.2.2 PMX Tools

Similarly to PBPK, PMX knows a much smaller tools selection than the SB community [72] **(**Fig. 2). This has multiple reasons. For once, the community is significantly smaller than that of computational biology around standards such as SBML and CellML. Missing exchange standards is another issue, but the major difficulty that tool designers face is the complexity of non-linear mixed effect (NLME) models and the demanding requirements with respect to the optimization algorithms vis-à-vis typical ODE-based models. A couple of software tools are available, such as Adapt II [73], Monolix [74], NONMEM [75]**,** Phoenix NLME [72] or WinBUGS [76] and we will describe two of them that represent different modeling approaches that are relevant for discussion.

The best-known tool in Pharmacometrics, **NONMEM**, has been developed in the early 1980s and is still under active development [75] (Table 1). Based on FORTRAN, it allows users to encode almost any model scenario in the imperative language NMTRAN (in contrast to other tools using declarative languages). This comes with the setback that it requires non-trivial programming skills from the modeler in order to encode standard statistical models. It has a built-in library of PK models equipped with a number of ready-to-go routines that cover a few common compartmental models, while other models have to be encoded using ODEs. The supported dataset is of the event-driven type and provides observation and dosing records, covariates and—implicitly—comprehensive information about the underlying trial design with its phases, occasions, resetting events, etc. A number of third-party tools are at user disposal, enabling not only a more convenient working environment with this command-based tool but also complex analysis of the estimation results [77, 78].

Monolix is a relatively new software tool that is quickly gaining popularity, in part due to the introduction of a novel and powerful estimation method [79] (Table 1). One major advantage is the declarative modeling language MLXTRAN, which has a well-defined vocabulary and grammar, and clear boundaries. It allows defining models in a highly structured manner, is accessible to a wide audience, and is easy to learn for beginners, which becomes clear when dealing with discrete data models. Monolix handles any number of variability levels, mixture models, below limit of quantification (BLQ) data, etc. It is equipped with a powerful GUI enabling the user to interact with the tool by setting the dataset and structural model of interest, covariate model, parameter distributions, and residual error model type. It can be used to set initial values for fixed effects, standard deviations for random effects, residual error model parameters, and numerical algorithms settings. Three other tools accompany Monolix: *simulx*—a simulator for clinical trials; *mlxplore*—software for the exploration and visualization of complex PMX models; and *datxplore*—a tool for the exploration and visualization of data.

The majority of tools used in pharmacometrics are available only commercially. Some of them however are coming with a free academic license (e.g. Simcyp Simulator, PK-Sim, or Monolix). Compared to the mainly academically developed open source SB tools, their code is usually closed.

5 Conclusion: Standards and Tools for Systems Medicine

In this chapter, we have broadly approached the development of tools for use in Systems Medicine from two different perspectives: the Systems Biology and SB-QSP-PMX domains. In both cases we have highlighted different strategies, used in the development of a number of tools, and showed how these affect the choice of software for a particular task (Fig. 2). In the case of simulation tools, the issues of user interface, end-user environment, and especially support for open standards are highlighted as being critical not only for data interoperability and reproducibility, but also for traceability. When considering the substantial change in role that models, and their instantiation in simulation software, must go through when moving from simply being research tools to, for example, being "software as medical devices" one sees how this could, inevitably, have a large effect on how future tools are developed, maintained, and supported. However, these issues are not unique, a point which becomes clear when the question of tool development is approached from a Systems Pharmacology perspective.

There exist many hundreds of tools in the combined SB-QSP-PMX area and there arises the question of how to coordinate and streamline the process using this plethora of tools. Model exchange among tools requires often error prone manual re-coding and data conversions and should be avoided whenever possible. What is needed is, first, an interoperability platform based on a set of exchange formats assuring the required compatibility of available tools. Second, a generic data format is needed covering all features expected from single subject datasets as used in SB or QSP and the population data needed to feed the PMX tools. With SBML and PharmML we have, in fact, come very close to achieving this goal—we have a set of complementary formats covering major model types used at any stage of the drug discovery and development process. The above-discussed SBW framework can serve as a working example for such a platform driven by the SBML standard.

In the end, no matter the approach, the lessons learned in both these domains, especially with respect to standardization and interoperability, will be invaluable in the upcoming and challenging task of designing tools for Systems Medicine that have a real benefit to society.

Notes

Acknowledgments

Brett Olivier is supported by a BE-Basic Foundation grant F08.005.001. Maciej Swat has received support from the Innovative Medicines Initiative Joint Undertaking under grant agreement n° 115156, resources of which are composed of financial contributions from the European Union's Seventh Framework Programme (FP7/2007-2013) and EFPIA companies' in kind contribution. The DDMoRe project is also supported by financial contribution from academic and SME partners.

References

1. Bruggeman FJ, Westerhoff HV (2007) The nature of systems biology. Trends Microbiol 15:45–50
CrossRef (https://doi.org/10.1016/j.tim.2006.11.003)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=17113776)
Google Scholar (http://scholar.google.com/scholar_lookup?title=The%20nature%20of%20systems%20biology&author=FJ.%20Bruggeman&author=HV.%20Westerhoff&journal=Trends%20Microbiol&volume=15&pages=45-50&publication_year=2007)

2. Garfinkel D, Garfinkel L, Pring M et al (1970) Computer applications to biochemical kinetics. Annu Rev Biochem 39:473–498
CrossRef (https://doi.org/10.1146/annurev.bi.39.070170.002353)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=4921695)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Computer%20applications%20to%20biochemical%20kinetics&author=D.%20Garfinkel&author=L.%20Garfinkel&author=M.%20Pring&journal=Annu%20Rev%20Biochem&volume=39&pages=473-498&publication_year=1970)

3. Savinell JM, Palsson BO (1992) Network analysis of intermediary metabolism using linear optimization. I Development of mathematical formalism. J Theor Biol 154:421–445
CrossRef (https://doi.org/10.1016/S0022-5193(05)80161-4)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=1593896)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Network%20analysis%20of%20intermediary%20metabolism%20using%20linear%20optimization.%20I%20Development%20of%20mathematical%20formalism&author=JM.%20Savinell&author=BO.%20Palsson&journal=J%20Theor%20Biol&volume=154&pages=421-445&publication_year=1992)

4. Hardy S, Robillard PN (2004) Modeling and simulation of molecular biology systems using petri nets: modeling goals of various approaches. J Bioinform Comput Biol 2:595–613
CrossRef (https://doi.org/10.1142/S0219720004000764)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=15617157)

Google Scholar (http://scholar.google.com/scholar_lookup?title=Modeling%20and%20simulation%20of%20molecular%20biology%20systems%20using%20petri%20nets%3A%20modeling%20goals%20of%20various%20approaches&author=S.%20Hardy&author=PN.%20Robillard&journal=J%20Bioinform%20Comput%20Biol&volume=2&pages=595-613&publication_year=2004)

5. Liepe J, Kirk P, Filippi S et al (2014) A framework for parameter estimation and model selection from experimental data in systems biology using approximate Bayesian computation. Nat Protoc 2:439–456
CrossRef (https://doi.org/10.1038/nprot.2014.025)
Google Scholar (http://scholar.google.com/scholar_lookup?title=A%20framework%20for%20parameter%20estimation%20and%20model%20selection%20from%20experimental%20data%20in%20systems%20biology%20using%20approximate%20Bayesian%20computation&author=J.%20Liepe&author=P.%20Kirk&author=S.%20Filippi&journal=Nat%20Protoc&volume=2&pages=439-456&publication_year=2014)

6. CASyM Consortium (2014) The CASyM roadmap: implementation of systems medicine, version 1.0.
https://www.casym.eu/lw_resource/datapool/_items/item_328/roadmap_1.0.pd (https://www.casym.eu/lw_resource/datapool/_items/item_328/roadmap_1.0.pdf). Accessed 4 Dec 2004

7. Flores M, Glusman G, Brogaard K et al (2014) P4 medicine: how systems medicine will transform the healthcare sector and society. Per Med 10:565–576
CrossRef (https://doi.org/10.2217/pme.13.57)
Google Scholar (http://scholar.google.com/scholar_lookup?title=P4%20medicine%3A%20how%20systems%20medicine%20will%20transform%20the%20healthcare%20sector%20and%20society&author=M.%20Flores&author=G.%20Glusman&author=K.%20Brogaard&journal=Per%20Med&volume=10&pages=565-576&publication_year=2014)

8. Heinrich R, Rapoport SM, Rapoport TA (1977) Metabolic regulation and mathematical models. Progr Biophys Mol Biol 32:1–82
CrossRef (https://doi.org/10.1016/0079-6107(78)90017-2)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Metabolic%20regulation%20and%20mathematical%20models&author=R.%20Heinrich&author=SM.%20Rapoport&author=TA.%20Rapoport&journal=Progr%20Biophys%20Mol%20Biol&volume=32&pages=1-82&publication_year=1977)

9. Wright BE, Kelly PJ (1981) Kinetic models of metabolism in intact cells, tissues and organisms. Curr Top Cell Regul 19:103–158
CrossRef (https://doi.org/10.1016/B978-0-12-152819-5.50021-X)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=7037313)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Kinetic%20models%20of%20metabolism%20in%20intact%20cells%2C%20tissues%20and%20organisms&author=BE.%20Wright&author=PJ.%20Kelly&journal=Curr%20Top%20Cell%20Regul&volume=19&pages=103-158&publication_year=1981)

10. Massoud TF, Hademenos GJ, Young WL et al (1998) Principles and philosophy of modeling in biomedical research. FASEB J 12:275–285

PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=9580086)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Principles%20and%20philosophy%20of%20modeling%20in%20biomedical%20research&author=TF.%20Massoud&author=GJ.%20Hademenos&author=WL.%20Young&journal=FASEB%20J&volume=12&pages=275-285&publication_year=1998)

11. Bakker BM et al (2010) Systems biology from micro-organisms to human metabolic diseases: the role of detailed kinetic models. Biochem Soc Trans 38:1294–1301
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3298031)
CrossRef (https://doi.org/10.1042/BST0381294)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=20863302)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Systems%20biology%20from%20micro-organisms%20to%20human%20metabolic%20diseases%3A%20the%20role%20of%20detailed%20kinetic%20models&author=BM.%20Bakker&journal=Biochem%20Soc%20Trans&volume=38&pages=1294-1301&publication_year=2010)

12. Orth JD et al (2010) What is flux balance analysis? Nat Biotechnol 28:245–248
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3108565)
CrossRef (https://doi.org/10.1038/nbt.1614)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=20212490)
Google Scholar (http://scholar.google.com/scholar_lookup?title=What%20is%20flux%20balance%20analysis%3F&author=JD.%20Orth&journal=Nat%20Biotechnol&volume=28&pages=245-248&publication_year=2010)

13. Thiele I, Swainston N, Fleming RMT et al (2013) A community-driven global reconstruction of human metabolism. Nat Biotechnol 31:419–425
CrossRef (https://doi.org/10.1038/nbt.2488)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=23455439)
Google Scholar (http://scholar.google.com/scholar_lookup?title=A%20community-driven%20global%20reconstruction%20of%20human%20metabolism&author=I.%20Thiele&author=N.%20Swainston&author=RMT.%20Fleming&journal=Nat%20Biotechnol&volume=31&pages=419-425&publication_year=2013)

14. Hoops S, Sahle S, Gauges R et al (2006) COPASI: a COmplex PAthway SImulator. Bioinformatics 22:3067–3074
CrossRef (https://doi.org/10.1093/bioinformatics/btl485)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=17032683)
Google Scholar (http://scholar.google.com/scholar_lookup?title=COPASI%3A%20a%20COmplex%20PAthway%20SImulator&author=S.%20Hoops&author=S.%20Sahle&author=R.%20Gauges&journal=Bioinformatics&volume=22&pages=3067-3074&publication_year=2006)

15. Sauro HM, Hucka M, Finney A et al (2003) Next generation simulation tools: the Systems Biology Workbench and BioSPICE integration. OMICS 7:355–372

CrossRef (https://doi.org/10.1089/153623103322637670)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=14683609)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Next%20generation%20simulation%20tools%3A%20the%20Systems%20Biology%20Workbench%20and%20BioSPICE%20integration&author=HM.%20Sauro&author=M.%20Hucka&author=A.%20Finney&journal=OMICS&volume=7&pages=355-372&publication_year=2003)

16. Olivier BG, Rohwer JM, Hofmeyr J-HS (2005) Modelling cellular systems with PySCeS. Bioinformatics 21:560–561
CrossRef (https://doi.org/10.1093/bioinformatics/bti046)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=15454409)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Modelling%20cellular%20systems%20with%20PySCeS&author=BG.%20Olivier&author=JM.%20Rohwer&author=J-HS.%20Hofmeyr&journal=Bioinformatics&volume=21&pages=560-561&publication_year=2005)

17. Sauro HM, Fell DA (1991) SCAMP: a metabolic simulator and control analysis program. Mathl Comput Model 15:15–28
CrossRef (https://doi.org/10.1016/0895-7177(91)90038-9)
Google Scholar (http://scholar.google.com/scholar_lookup?title=SCAMP%3A%20a%20metabolic%20simulator%20and%20control%20analysis%20program&author=HM.%20Sauro&author=DA.%20Fell&journal=Mathl%20Comput%20Model&volume=15&pages=15-28&publication_year=1991)

18. Klamt S, Saez-Rodriguez J, Gilles ED (2007) Structural and functional analysis of cellular networks with Cell NetAnalyzer. BMC Syst Biol 1:2
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1847467)
CrossRef (https://doi.org/10.1186/1752-0509-1-2)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=17408509)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Structural%20and%20functional%20analysis%20of%20cellular%20networks%20with%20Cell%20NetAnalyzer&author=S.%20Klamt&author=J.%20Saez-Rodriguez&author=ED.%20Gilles&journal=BMC%20Syst%20Biol&volume=1&pages=2&publication_year=2007)

19. Oberhardt MA, Palsson BØ, Papin JA (2009) Applications of genome-scale metabolic reconstructions. Mol Syst Biol 5:320
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2795471)
CrossRef (https://doi.org/10.1038/msb.2009.77)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=19888215)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Applications%20of%20genome-scale%20metabolic%20reconstructions&author=MA.%20Oberhardt&author=B%C3%98.%20Palsson&author=JA.%20Papin&journal=Mol%20Syst%20Biol&volume=5&pages=320&publication_year=2009)

20. Li C, Donizelli M, Rodriguez N et al (2010) BioModels Database: an enhanced, curated and annotated resource for published quantitative kinetic models. BMC Syst Biol 4:92
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2909940)
CrossRef (https://doi.org/10.1186/1752-0509-4-92)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=20587024)
Google Scholar (http://scholar.google.com/scholar_lookup?title=BioModels%20Database%3A%20an%20enhanced%2C%20curated%20and%20annotated%20resource%20for%20published%20quantitative%20kinetic%20models&author=C.%20Li&author=M.%20Donizelli&author=N.%20Rodriguez&journal=BMC%20Syst%20Biol&volume=4&pages=92&publication_year=2010)

21. Le Novère N, Bornstein B, Broicher A et al (2006) BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. Nucleic Acids Res 34:D689–D691
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1347454)
CrossRef (https://doi.org/10.1093/nar/gkj092)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=16381960)
Google Scholar (http://scholar.google.com/scholar_lookup?title=BioModels%20Database%3A%20a%20free%2C%20centralized%20database%20of%20curated%2C%20published%2C%20quantitative%20kinetic%20models%20of%20biochemical%20and%20cellular%20systems&author=N.%20Nov%C3%A8re&author=B.%20Bornstein&author=A.%20Broicher&journal=Nucleic%20Acids%20Res&volume=34&pages=D689-D691&publication_year=2006)

22. Benson DA, Karsch-Mizrachi I, Lipman DJ et al (2011) GenBank. Nucleic Acids Res 39:D32–D37
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3013681)
CrossRef (https://doi.org/10.1093/nar/gkq1079)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=21071399)
Google Scholar (http://scholar.google.com/scholar_lookup?title=GenBank&author=DA.%20Benson&author=I.%20Karsch-Mizrachi&author=DJ.%20Lipman&journal=Nucleic%20Acids%20Res&volume=39&pages=D32-D37&publication_year=2011)

23. Wimalaratne SM, Grenon P, Hermjakob H et al (2014) BioModels linked dataset. BMC Syst Biol 8:91
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4423647)
CrossRef (https://doi.org/10.1186/s12918-014-0091-5)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=25182954)
Google Scholar (http://scholar.google.com/scholar_lookup?title=BioModels%20linked%20dataset&author=SM.%20Wimalaratne&author=P.%20Grenon&author=H.%20Hermjakob&journal=BMC%20Syst%20Biol&volume=8&pages=91&publication_year=2014)

24. Olivier BG, Rohwer JM, Hofmeyr J-HS (2002) Modelling cellular processes with Python and SciPy. Mol Biol Rep 29:249–254
CrossRef (https://doi.org/10.1023/A%3A1020346417223)

PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=12241066)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Modelling%20cellular%20processes%20with%20Python%20and%20SciPy&author=BG.%20Olivier&author=JM.%20Rohwer&author=J-HS.%20Hofmeyr&journal=Mol%20Biol%20Rep&volume=29&pages=249-254&publication_year=2002)

25. Shapiro BE, Hucka M, Finney A et al (2004) MathSBML: a package for manipulating SBML-based biological models. Bioinformatics 20:2829–2831
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1409765)
CrossRef (https://doi.org/10.1093/bioinformatics/bth271)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=15087311)
Google Scholar (http://scholar.google.com/scholar_lookup?title=MathSBML%3A%20a%20package%20for%20manipulating%20SBML-based%20biological%20models&author=BE.%20Shapiro&author=M.%20Hucka&author=A.%20Finney&journal=Bioinformatics&volume=20&pages=2829-2831&publication_year=2004)

26. Schellenberger J, Que R, Fleming RMT et al (2011) Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0. Nat Protoc 6:1290–1307
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3319681)
CrossRef (https://doi.org/10.1038/nprot.2011.308)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=21886097)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Quantitative%20prediction%20of%20cellular%20metabolism%20with%20constraint-based%20models%3A%20the%20COBRA%20Toolbox%20v2.0&author=J.%20Schellenberger&author=R.%20Que&author=RMT.%20Fleming&journal=Nat%20Protoc&volume=6&pages=1290-1307&publication_year=2011)

27. Olivier BG, Snoep JL (2004) Web-based kinetic modelling using JWS Online. Bioinformatics 20:2143–2144
CrossRef (https://doi.org/10.1093/bioinformatics/bth200)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=15072998)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Web-based%20kinetic%20modelling%20using%20JWS%20Online&author=BG.%20Olivier&author=JL.%20Snoep&journal=Bioinformatics&volume=20&pages=2143-2144&publication_year=2004)

28. IBM Corporation (2014) IBM ILOG CPLEX optimizer. http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer (http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer). Accessed 29 Nov 2014

29. Gurobi Optimization, Inc. (2014) Gurobi optimizer reference manual. http://www.gurobi.com (http://www.gurobi.com/). Accessed 27 Nov 2014

30. Hucka M, Finney A (2005) Escalating model sizes and complexities call for standardized forms of representation. Mol Syst Biol 1:2005.0011

PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1360139)
CrossRef (https://doi.org/10.1038/msb4100015)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=16729046)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Escalating%20model%20sizes%20and%20complexities%20call%20for%20standardized%20forms%20of%20representation&author=M.%20Hucka&author=A.%20Finney&journal=Mol%20Syst%20Biol&volume=1&pages=2005.0011&publication_year=2005)

31. Keating SM, Le Novère N (2013) Supporting SBML as a model exchange format in software applications. Methods Mol Biol 1021:201–225
CrossRef (https://doi.org/10.1007/978-1-62703-450-0_11)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=23715987)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Supporting%20SBML%20as%20a%20model%20exchange%20format%20in%20software%20applications.&author=SM.%20Keating&author=N.%20Nov%C3%A8re&journal=Methods%20Mol%20Biol&volume=1021&pages=201-225&publication_year=2013)

32. Waltemath D, Bergmann FT, Chaouiya C et al (2014) Meeting report from the fourth meeting of the Computational Modeling in Biology Network (COMBINE). Stand Genomic Sci 9:3
CrossRef (https://doi.org/10.4056/sigs.5279417)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Meeting%20report%20from%20the%20fourth%20meeting%20of%20the%20Computational%20Modeling%20in%20Biology%20Network%20%28COMBINE%29&author=D.%20Waltemath&author=FT.%20Bergmann&author=C.%20Chaouiya&journal=Stand%20Genomic%20Sci&volume=9&pages=3&publication_year=2014)

33. Miller AK, Marsh J, Reeve A et al (2010) An overview of the CellML API and its implementation. BMC Bioinformatics 11:178
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2858041)
CrossRef (https://doi.org/10.1186/1471-2105-11-178)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=20377909)
Google Scholar (http://scholar.google.com/scholar_lookup?title=An%20overview%20of%20the%20CellML%20API%20and%20its%20implementation&author=AK.%20Miller&author=J.%20Marsh&author=A.%20Reeve&journal=BMC%20Bioinformatics&volume=11&pages=178&publication_year=2010)

34. Galdzicki M, Clancy KP, Oberortner E et al (2014) The Synthetic Biology Open Language (SBOL) provides a community standard for communicating designs in synthetic biology. Nat Biotechnol 32:545–550
CrossRef (https://doi.org/10.1038/nbt.2891)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=24911500)
Google Scholar (http://scholar.google.com/scholar_lookup?title=The%20Synthetic%20Biology%20Open%20Language%20%28SBOL%29%2

0provides%20a%20community%20standard%20for%20communicating%20designs%20in%20synthetic%20biology&author=M.%20Galdzicki&author=KP.%20Clancy&author=E.%20Oberortner&journal=Nat%20Biotechnol&volume=32&pages=545-550&publication_year=2014)

35. Le Novère N, Finney A, Hucka M et al (2005) Minimum information requested in the annotation of biochemical models (MIRIAM). Nat Biotechnol 23:1509–1515
CrossRef (https://doi.org/10.1038/nbt1156)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=16333295)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Minimum%20information%20requested%20in%20the%20annotation%20of%20biochemical%20models%20%28MIRIAM%29&author=N.%20Nov%C3%A8re&author=A.%20Finney&author=M.%20Hucka&journal=Nat%20Biotechnol&volume=23&pages=1509-1515&publication_year=2005)

36. Waltemath D, Adams R, Beard DA et al (2011) Minimum Information About a Simulation Experiment (MIASE). PLoS Comput Biol 7:e1001122
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3084216)
CrossRef (https://doi.org/10.1371/journal.pcbi.1001122)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=21552546)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Minimum%20Information%20About%20a%20Simulation%20Experiment%20%28MIASE%29&author=D.%20Waltemath&author=R.%20Adams&author=DA.%20Beard&journal=PLoS%20Comput%20Biol&volume=7&pages=e1001122&publication_year=2011)

37. Hucka M, Finney A, Sauro HM et al (2003) The Systems Biology Markup Language (SBML): A medium for representation and exchange of biochemical network models. Bioinformatics 9:524–531
CrossRef (https://doi.org/10.1093/bioinformatics/btg015)
Google Scholar (http://scholar.google.com/scholar_lookup?title=The%20Systems%20Biology%20Markup%20Language%20%28SBML%29%3A%20A%20medium%20for%20representation%20and%20exchange%20of%20biochemical%20network%20models&author=M.%20Hucka&author=A.%20Finney&author=HM.%20Sauro&journal=Bioinformatics&volume=9&pages=524-531&publication_year=2003)

38. Laibe C, Le Novère N (2007) MIRIAM Resources: tools to generate and resolve robust cross-references in systems biology. BMC Syst Biol 1:58
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2259379)
CrossRef (https://doi.org/10.1186/1752-0509-1-58)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=18078503)
Google Scholar (http://scholar.google.com/scholar_lookup?title=MIRIAM%20Resources%3A%20tools%20to%20generate%20and%20resolve%20robust%20cross-references%20in%20systems%20biology&author=C.%20Laibe&author=N.%20Nov%C3%A8re&journal=BMC%20Syst%20Biol&volume=1&pages=58&publication_year=2007)

39. Juty N, Le Novère N, Laibe C (2012) Identifiers.org and MIRIAM Registry: community resources to provide persistent identification. Nucleic Acids Res 40:D580–D586
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3245029)
CrossRef (https://doi.org/10.1093/nar/gkr1097)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=22140103)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Identifiers.org%20and%20MIRIAM%20Registry%3A%20community%20resources%20to%20provide%20persistent%20identification&author=N.%20Juty&author=N.%20Nov%C3%A8re&author=C.%20Laibe&journal=Nucleic%20Acids%20Res&volume=40&pages=D580-D586&publication_year=2012)

40. Olivier, BG, Bergmann, FT (2015) The Systems Biology Markup Language (SBML) Level 3 Package: Flux Balance Constraints. Journal of Integrative Bioinformatics, 12:269
Google Scholar (https://scholar.google.com/scholar?q=Olivier%2C%20BG%2C%20Bergmann%2C%20FT%20%282015%29%20The%20Systems%20Biology%20Markup%20Language%20%28SBML%29%20Level%203%20Package%3A%20Flux%20Balance%20Constraints.%20Journal%20of%20Integrative%20Bioinformatics%2C%2012%3A269)

41. Chaouiya C, Berenguier D, Keating SM et al (2013) SBML Qualitative Models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools. BMC Syst Biol 7:135
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3892043)
CrossRef (https://doi.org/10.1186/1752-0509-7-135)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=24321545)
Google Scholar (http://scholar.google.com/scholar_lookup?title=SBML%20Qualitative%20Models%3A%20a%20model%20representation%20format%20and%20infrastructure%20to%20foster%20interactions%20between%20qualitative%20modelling%20formalisms%20and%20tools&author=C.%20Chaouiya&author=D.%20Berenguier&author=SM.%20Keating&journal=BMC%20Syst%20Biol&volume=7&pages=135&publication_year=2013)

42. Bornstein BJ, Keating SM, Jouraku A et al (2008) LibSBML: an API library for SBML. Bioinformatics 24:880–881
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2517632)
CrossRef (https://doi.org/10.1093/bioinformatics/btn051)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=18252737)
Google Scholar (http://scholar.google.com/scholar_lookup?title=LibSBML%3A%20an%20API%20library%20for%20SBML&author=BJ.%20Bornstein&author=SM.%20Keating&author=A.%20Jouraku&journal=Bioinformatics&volume=24&pages=880-881&publication_year=2008)

43. Dräger A, Rodriguez N, Dumousseau M et al (2011) JSBML: a flexible Java library for working with SBML. Bioinformatics 27:2167–2168
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3137227)
CrossRef (https://doi.org/10.1093/bioinformatics/btr361)

PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=21697129)
Google Scholar (http://scholar.google.com/scholar_lookup?title=JSBML%3A%20a%20flexible%20Java%20library%20for%20working%20with%20SBML&author=A.%20Dr%C3%A4ger&author=N.%20Rodriguez&author=M.%20Dumousseau&journal=Bioinformatics&volume=27&pages=2167-2168&publication_year=2011)

44. Waltemath D, Adams R, Bergmann FT et al (2011) Reproducible computational biology experiments with SED-ML – The Simulation Experiment Description Markup Language. BMC Syst Biol 5:198
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3292844)
CrossRef (https://doi.org/10.1186/1752-0509-5-198)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=22172142)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Reproducible%20computational%20biology%20experiments%20with%20SED-ML%20%E2%80%93%20The%20Simulation%20Experiment%20Description%20Markup%20Language&author=D.%20Waltemath&author=R.%20Adams&author=FT.%20Bergmann&journal=BMC%20Syst%20Biol&volume=5&pages=198&publication_year=2011)

45. IMDRF SaMD Working Group (2013) Software as a Medical Device (SaMD): key definitions. http://www.imdrf.org/documents/documents.asp (http://www.imdrf.org/documents/documents.asp). Accessed 25 Nov 2014

46. Iec I (2006) 62304: 2006 Medical device software – software life cycle processes. International Electrotechnical Commission, Geneva
Google Scholar (http://scholar.google.com/scholar_lookup?title=62304%3A%202006%20Medical%20device%20software%20%E2%80%93%20software%20life%20cycle%20processes&author=I.%20Iec&publication_year=2006)

47. Buntz B (2011) Simplifying IEC 62304 compliance for developers. http://www.emdt.co.uk/article/iec-62304-compliance?utm_source=emdt&utm_medium=articlebottom&utm_campaign=camilla (http://www.emdt.co.uk/article/iec-62304-compliance?utm_source=emdt%26utm_medium=articlebottom%26utm_campaign=camilla). Accessed 25 Nov 2014

48. Regan G, McCaffery F, McDaid K et al (2013) Medical device standards' requirements for traceability during the software development lifecycle and implementation of a traceability assessment model. Comput Stand Int 36:3–9
CrossRef (https://doi.org/10.1016/j.csi.2013.07.012)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Medical%20device%20standards%E2%80%99%20requirements%20for%20traceability%20during%20the%20software%20development%20lifecycle%20and%20implementation%20of%20a%20traceability%20assessment%20model&author=G.%20Regan&author=F.%20McCaffery&author=K.%20McDaid&journal=Comput%20Stand%20Int&volume=36&pages=3-9&publication_year=2013)

49. Gotel OCZ, Finkelstein CW (1994) An analysis of the requirements traceability problem. Proceedings of the First International Conference on Requirements

Engineering. pp 94–101
Google Scholar (https://scholar.google.com/scholar?q=Gotel%20OCZ%2C%20Finkelstein%20CW%20%281994%29%20An%20analysis%20of%20the%20requirements%20traceability%20problem.%20Proceedings%20of%20the%20First%20International%20Conference%20on%20Requirements%20Engineering.%20pp%2094%E2%80%93101)

50. Lakshmanan M, Koh G, Chung BKS et al (2014) Software applications for flux balance analysis. Brief Bioinform 15:108–122
CrossRef (https://doi.org/10.1093/bib/bbs069)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=23131418)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Software%20applications%20for%20flux%20balance%20analysis&author=M.%20Lakshmanan&author=G.%20Koh&author=BKS.%20Chung&journal=Brief%20Bioinform&volume=15&pages=108-122&publication_year=2014)

51. Kent E, Hoops S, Mendes P (2012) Condor-COPASI: high-throughput computing for biochemical networks. BMC Syst Biol 6:91
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3527284)
CrossRef (https://doi.org/10.1186/1752-0509-6-91)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=22834945)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Condor-COPASI%3A%20high-throughput%20computing%20for%20biochemical%20networks&author=E.%20Kent&author=S.%20Hoops&author=P.%20Mendes&journal=BMC%20Syst%20Biol&volume=6&pages=91&publication_year=2012)

52. Ebrahim A, Lerman JA, Palsson BO et al (2013) COBRApy: Constraints-Based Reconstruction and Analysis for Python. BMC Syst Biol 7:74
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3751080)
CrossRef (https://doi.org/10.1186/1752-0509-7-74)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=23927696)
Google Scholar (http://scholar.google.com/scholar_lookup?title=COBRApy%3A%20Constraints-Based%20Reconstruction%20and%20Analysis%20for%20Python&author=A.%20Ebrahim&author=JA.%20Lerman&author=BO.%20Palsson&journal=BMC%20Syst%20Biol&volume=7&pages=74&publication_year=2013)

53. Bergmann FT, Vallabhajosyula RR, Sauro HM (2006) Computational tools for modeling protein networks. Curr Proteomics 3:181–197
CrossRef (https://doi.org/10.2174/157016406779475380)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Computational%20tools%20for%20modeling%20protein%20networks&author=FT.%20Bergmann&author=RR.%20Vallabhajosyula&author=HM.%20Sauro&journal=Curr%20Proteomics&volume=3&pages=181-197&publication_year=2006)

54. Hucka M, Finney A, Sauro HM et al. (2002) The ERATO Systems Biology Workbench: enabling interaction and exchange between software tools for computational biology. Pac Symp Biocomput 450–461

Google Scholar (https://scholar.google.com/scholar?q=Hucka%20M%2C%20Finney%20A%2C%20Sauro%20HM%20et%20al.%20%282002%29%20The%20ERATO%20Systems%20Biology%20Workbench%3A%20enabling%20interaction%20and%20exchange%20between%20software%20tools%20for%20computational%20biology.%20Pac%20Symp%20Biocomput%20450%E2%80%93461)

55. Smith LP, Bergmann FT, Chandran D et al (2009) Antimony: a modular model definition language. Bioinformatics 25:2452–2454
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2735663)
CrossRef (https://doi.org/10.1093/bioinformatics/btp401)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=19578039)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Antimony%3A%20a%20modular%20model%20definition%20language&author=LP.%20Smith&author=FT.%20Bergmann&author=D.%20Chandran&journal=Bioinformatics&volume=25&pages=2452-2454&publication_year=2009)

56. Boele J, Olivier BG, Teusink B (2012) FAME, the Flux Analysis and Modeling Environment. BMC Syst Biol 6:8
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3317868)
CrossRef (https://doi.org/10.1186/1752-0509-6-8)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=22289213)
Google Scholar (http://scholar.google.com/scholar_lookup?title=FAME%2C%20the%20Flux%20Analysis%20and%20Modeling%20Environment&author=J.%20Boele&author=BG.%20Olivier&author=B.%20Teusink&journal=BMC%20Syst%20Biol&volume=6&pages=8&publication_year=2012)

57. Kanehisa M, Goto S (2000) KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Res 28:27–30
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC102409)
CrossRef (https://doi.org/10.1093/nar/28.1.27)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=10592173)
Google Scholar (http://scholar.google.com/scholar_lookup?title=KEGG%3A%20Kyoto%20Encyclopedia%20of%20Genes%20and%20Genomes&author=M.%20Kanehisa&author=S.%20Goto&journal=Nucleic%20Acids%20Res&volume=28&pages=27-30&publication_year=2000)

58. Byon W, Smith MK, Chan P et al (2013) Establishing best practices and guidance in population modeling: an experience with an internal population pharmacokinetic analysis guidance. CPT Pharmacometrics Syst Pharmacol 2:e51
CrossRef (https://doi.org/10.1038/psp.2013.26)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=23836283)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Establishing%20best%20practices%20and%20guidance%20in%20population%20modeling%3A%20an%20experience%20with%20an%20internal%20population%20pharmacokinetic%20analysis%20guidance&author=W.%20Byon&author=MK.%20Smith&author=P.%20Chan&journal=CPT%20Pharmacometrics%20Syst%20Pharmacol&volume=2&pages=e51&publication_year=2013)

59. Agoram BM, Demin O (2011) Integration not isolation: arguing the case for quantitative and systems pharmacology in drug discovery and development. Drug Discovery Today 16(23–24)
Google Scholar (https://scholar.google.com/scholar?q=Agoram%20BM%2C%20Demin%20O%20%282011%29%20Integration%20not%20isolation%3A%20arguing%20the%20case%20for%20quantitative%20and%20systems%20pharmacology%20in%20drug%20discovery%20and%20development.%20Drug%20Discovery%20Today%2016%2823%E2%80%9324%29)

60. Mager D, Jusko W (2008) Development of translational pharmacokinetic-pharmacodynamic models. Clin Pharmacol Ther 83(6):909–912
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2671003)
CrossRef (https://doi.org/10.1038/clpt.2008.52)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=18388873)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Development%20of%20translational%20pharmacokinetic-%20pharmacodynamic%20models&author=D.%20Mager&author=W.%20Jusko&journal=Clin%20Pharmacol%20Ther&volume=83&issue=6&pages=909-912&publication_year=2008)

61. Rostami-Hodjegan A, Tucker GT (2007) Simulation and prediction of in vivo drug metabolism in human populations from in vitro data. Nat Rev Drug Discov 6:140–148
CrossRef (https://doi.org/10.1038/nrd2173)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=17268485)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Simulation%20and%20prediction%20of%20in%20vivo%20drug%20metabolism%20in%20human%20populations%20from%20in%20vitro%20data&author=A.%20Rostami-Hodjegan&author=GT.%20Tucker&journal=Nat%20Rev%20Drug%20Discov&volume=6&pages=140-148&publication_year=2007)

62. Jones HM, Rowland-Yeo K (2013) Basic concepts in physiologically based pharmacokinetic modeling in drug discovery and development. CPT Pharmacometrics Syst Pharmacol 2:e63
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3828005)
CrossRef (https://doi.org/10.1038/psp.2013.41)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=23945604)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Basic%20concepts%20in%20physiologically%20based%20pharmacokinetic%20modeling%20in%20drug%20discovery%20and%20development&author=HM.%20Jones&author=K.%20Rowland-Yeo&journal=CPT%20Pharmacometrics%20Syst%20Pharmacol&volume=2&pages=e63&publication_year=2013)

63. Bonate P (2011) Pharmacokinetic-pharmacodynamic modeling and simulation, 2nd edn. Springer, New York
CrossRef (https://doi.org/10.1007/978-1-4419-9485-1)

Google Scholar (http://scholar.google.com/scholar_lookup?title=Pharmacokinetic-pharmacodynamic%20modeling%20and%20simulation&author=P.%20Bonate&publication_year=2011)

64. Lavielle M (2014) Mixed effects models for the population approach models, tasks, methods & tools, CRC biostatistics series. Chapman & Hall, Boca Raton, FL
Google Scholar (http://scholar.google.com/scholar_lookup?title=Mixed%20effects%20models%20for%20the%20population%20approach%20models%2C%20tasks%2C%20methods%20%26%20tools&author=M.%20Lavielle&publication_year=2014)

65. Leil TA, Bertz R (2014) Quantitative systems pharmacology can reduce attrition and improve productivity in pharmaceutical research and development. Front Pharmacol 5:247
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4226160)
CrossRef (https://doi.org/10.3389/fphar.2014.00247)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=25426074)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Quantitative%20systems%20pharmacology%20can%20reduce%20attrition%20and%20improve%20productivity%20in%20pharmaceutical%20research%20and%20development&author=TA.%20Leil&author=R.%20Bertz&journal=Front%20Pharmacol&volume=5&pages=247&publication_year=2014)

66. Kola I, Landis J (2004) Can the pharmaceutical industry reduce attrition rates? Nat Rev Drug Discov 3:711–716
CrossRef (https://doi.org/10.1038/nrd1470)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=15286737)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Can%20the%20pharmaceutical%20industry%20reduce%20attrition%20rates%3F&author=I.%20Kola&author=J.%20Landis&journal=Nat%20Rev%20Drug%20Discov&volume=3&pages=711-716&publication_year=2004)

67. Bazzoli C, Retout S, Mentré F (2010) Design evaluation and optimization in multiple response nonlinear mixed effect models: PFIM 3.0. Comput Methods Prog Biomed 98:55–65
CrossRef (https://doi.org/10.1016/j.cmpb.2009.09.012)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Design%20evaluation%20and%20optimization%20in%20multiple%20response%20nonlinear%20mixed%20effect%20models%3A%20PFIM%203.0&author=C.%20Bazzoli&author=S.%20Retout&author=F.%20Mentr%C3%A9&journal=Comput%20Methods%20Prog%20Biomed&volume=98&pages=55-65&publication_year=2010)

68. Draeger A, Palsson BO (2014) Improving collaboration by standardization efforts in systems biology. Front Bioeng Biotechnol. doi: 10.3389/fbioe.2014.00061 (https://doi.org/10.3389/fbioe.2014.00061)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Improving%20collaboration%20by%20standardization%20efforts%20in%20systems%20biology&author=A.%20Draeger&author=BO.%20Palsson&journal=

Front%20Bioeng%20Biotechnol&publication_year=2014&doi=10.3389%2Ffbioe.2014.00061)

69. Swat MJ (2015) Pharmacometrics Markup Language (PharmML): opening new perspectives for model exchange in drug development. CPT Pharmacometrics Syst Pharmacol 4(6):316–319. doi: 10.1002/psp4.57 (https://doi.org/10.1002/psp4.57)
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4505825)
CrossRef (https://doi.org/10.1002/psp4.57)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=26225259)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Pharmacometrics%20Markup%20Language%20%28PharmML%29%3A%20opening%20new%20perspectives%20for%20model%20exchange%20in%20drug%20development&author=MJ.%20Swat&journal=CPT%20Pharmacometrics%20Syst%20Pharmacol&volume=4&issue=6&pages=316-319&publication_year=2015&doi=10.1002%2Fpsp4.57)

70. Marciniak J (2009) The Simcyp population-based ADME simulator. Expert Opin Drug Metab Toxicol 5:211–223
CrossRef (https://doi.org/10.1517/17425250802691074)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=19199378)
Google Scholar (http://scholar.google.com/scholar_lookup?title=The%20Simcyp%20population-based%20ADME%20simulator&author=J.%20Marciniak&journal=Expert%20Opin%20Drug%20Metab%20Toxicol&volume=5&pages=211-223&publication_year=2009)

71. Eissing T, Kuepfer L, Becker C et al (2011) A computational systems biology software platform for multiscale modeling and simulation: integrating whole-body physiology, disease biology, and molecular reaction networks. Front Physiol 2:4
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3070480)
CrossRef (https://doi.org/10.3389/fphys.2011.00004)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=21483730)
Google Scholar (http://scholar.google.com/scholar_lookup?title=A%20computational%20systems%20biology%20software%20platform%20for%20multiscale%20modeling%20and%20simulation%3A%20integrating%20whole-body%20physiology%2C%20disease%20biology%2C%20and%20molecular%20reaction%20networks&author=T.%20Eissing&author=L.%20Kuepfer&author=C.%20Becker&journal=Front%20Physiol&volume=2&pages=4&publication_year=2011)

72. Aarons L (1999) Software for population pharmacokinetics and pharmacodynamics. Clin Pharmacokinet 36:255–264
CrossRef (https://doi.org/10.2165/00003088-199936040-00001)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=10320948)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Software%20for%20population%20pharmacokinetics%20and%20pharmaco

dynamics&author=L.%20Aarons&journal=Clin%20Pharmacokinet&volume=36&pages=255-264&publication_year=1999)

73. D'Argenio DZ, Schumitzky A, Wang X (2009) Adapt 5 user's guide: Pharmacokinetic/pharmacodynamic systems analysis software. Tech. Rep., Biomedical Simulations Resource, Los Angeles
Google Scholar (https://scholar.google.com/scholar?q=D%E2%80%99Argenio%20DZ%2C%20Schumitzky%20A%2C%20Wang%20X%20%282009%29%20Adapt%205%20user%27s%20guide%3A%20Pharmacokinetic%2Fpharmacodynamic%20systems%20analysis%20software.%20Tech.%20Rep.%2C%20Biomedical%20Simulations%20Resource%2C%20Los%20Angeles)

74. Lixoft. Monolix 4.3. http://lixoft.com (http://lixoft.com/)

75. Beal SL, Sheiner LB, Boeckmann AJ et al. (2009) NONMEM User's guides. Technical report. Icon Development Solutions, Ellicott City, MD, USA
Google Scholar (https://scholar.google.com/scholar?q=Beal%20SL%2C%20Sheiner%20LB%2C%20Boeckmann%20AJ%20et%20al.%20%282009%29%20NONMEM%20User%E2%80%99s%20guides.%20Technical%20report.%20Icon%20Development%20Solutions%2C%20Ellicott%20City%2C%20MD%2C%20USA)

76. Lunn DJ, Best N, Thomas A et al (2002) Bayesian analysis of population pk/pd models: general concepts and software. J Pharmacokinet Pharmacodyn 29:271–307
CrossRef (https://doi.org/10.1023/A%3A1020206907668)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=12449499)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Bayesian%20analysis%20of%20population%20pk%2Fpd%20models%3A%20general%20concepts%20and%20software&author=DJ.%20Lunn&author=N.%20Best&author=A.%20Thomas&journal=J%20Pharmacokinet%20Pharmacodyn&volume=29&pages=271-307&publication_year=2002)

77. Lindbom L, Ribbing J, Jonsson EN (2004) Perl–speaks–NONMEM (PsN) – a Perl module for NONMEM related programming. Comput Methods Programs Biomed 75(2):85–94
CrossRef (https://doi.org/10.1016/j.cmpb.2003.11.003)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=15212851)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Perl%E2%80%93speaks%E2%80%93NONMEM%20%28PsN%29%20%E2%80%93%20a%20Perl%20module%20for%20NONMEM%20related%20programming&author=L.%20Lindbom&author=J.%20Ribbing&author=EN.%20Jonsson&journal=Comput%20Methods%20Programs%20Biomed&volume=75&issue=2&pages=85-94&publication_year=2004)

78. Keizer R, Karlsson M, Hooker A (2013) Modeling and Simulation Workbench for NONMEM: tutorial on Pirana, PsN, and Xpose. CPT Pharmacometrics Syst Pharmacol 2:e50
PubMedCentral (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3697037)
CrossRef (https://doi.org/10.1038/psp.2013.24)
PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Abstract&list_uids=23836189)

Google Scholar (http://scholar.google.com/scholar_lookup?title=Modeling%20and%20Simulation%20Workbench%20for%20NONMEM%3A%20tutorial%20on%20Pirana%2C%20PsN%2C%20and%20Xpose&author=R.%20Keizer&author=M.%20Karlsson&author=A.%20Hooker&journal=CPT%20Pharmacometrics%20Syst%20Pharmacol&volume=2&pages=e50&publication_year=2013)

79. Kuhn E, Lavielle M (2005) Maximum likelihood estimation in nonlinear mixed effects models. Comput Stat Data Anal 49:1020–1038
CrossRef (https://doi.org/10.1016/j.csda.2004.07.002)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Maximum%20likelihood%20estimation%20in%20nonlinear%20mixed%20effects%20models&author=E.%20Kuhn&author=M.%20Lavielle&journal=Comput%20Stat%20Data%20Anal&volume=49&pages=1020-1038&publication_year=2005)

Copyright information

About this protocol

Cite this protocol as:

Olivier B.G., Swat M.J., Moné M.J. (2016) Modeling and Simulation Tools: From Systems Biology to Systems Medicine. In: Schmitz U., Wolkenhauer O. (eds) Systems Medicine. Methods in Molecular Biology, vol 1386. Humana Press, New York, NY

- DOI https://doi.org/10.1007/978-1-4939-3283-2_19
- Publisher Name Humana Press, New York, NY
- Print ISBN 978-1-4939-3282-5
- Online ISBN 978-1-4939-3283-2
- eBook Packages Springer Protocols

Personalised recommendations