

BioUML: an integrated environment for systems biology and collaborative analysis of biomedical data

Fedor Kolpakov^{1,2,*}, Ilya Akberdin^{1,3}, Timur Kashapov¹, Ilya Kiselev^{1,2}, Semyon Kolmykov^{1,2,4}, Yuri Kondrakhin^{1,2}, Elena Kutumova^{1,2}, Nikita Mandrik^{1,2}, Sergey Pintus^{1,2}, Anna Ryabova^{1,2}, Ruslan Sharipov^{1,3}, Ivan Yevshin^{1,2} and Alexander Kel^{1,5,6}

¹BIOFON.RU, LLC, Novosibirsk 630090, Russian Federation, ²Institute of Computational Technologies SB RAS, Novosibirsk 630090, Russian Federation, ³Novosibirsk State University, Novosibirsk 630090, Russian Federation, ⁴Institute of Cytology and Genetics SB RAS, Novosibirsk 630090, Russian Federation, ⁵geneXplain GmbH, 38302 Wolfenbüttel, Germany and ⁶Institute of Chemical Biology and Fundamental Medicine, SB RAS, Novosibirsk 630090, Russian Federation

Received March 13, 2019; Revised May 02, 2019; Editorial Decision May 07, 2019; Accepted May 11, 2019

ABSTRACT

BioUML (homepage: <http://www.biouml.org>, main public server: <https://ict.biouml.org>) is a web-based integrated environment (platform) for systems biology and the analysis of biomedical data generated by omics technologies. The BioUML vision is to provide a computational platform to build virtual cell, virtual physiological human and virtual patient. BioUML spans a comprehensive range of capabilities, including access to biological databases, powerful tools for systems biology (visual modelling, simulation, parameters fitting and analyses), a genome browser, scripting (R, JavaScript) and a workflow engine. Due to integration with the Galaxy platform and R/Bioconductor, BioUML provides powerful possibilities for the analyses of omics data. The plug-in-based architecture allows the user to add new functionalities using plug-ins. To facilitate a user focus on a particular task or database, we have developed several predefined perspectives that display only those web interface elements that are needed for a specific task. To support collaborative work on scientific projects, there is a central authentication and authorization system (<https://bio-store.org>). The diagram editor enables several remote users to simultaneously edit diagrams.

INTRODUCTION

The BioUML project was started in 2002, and its main goal was to develop a common purpose visual language for formal descriptions of the structure and function of biological

systems—a Biological Universal Modelling Language (BioUML) (1). As a starting point, we used the graphic notation suggested by the GeneNet system (2,3). We have developed several diagram types that allow the user to construct a biological model step-by-step, with increasing levels of detail and formality. Through this approach, we have developed two databases: Biopath (4) and Cyclonet (5).

Subsequently, the international community created the Systems Biology Graphical Notation (SBGN), which standardizes the graphical notation used in maps of biological processes (6). The BioUML team was involved in this standardization process. Currently, BioUML completely supports the SBGN Process Description diagrams.

From the beginning, BioUML has supported a paradigm of visual modelling, where the user can create a diagram that completely and formally specifies the given biological model. BioUML then automatically generates program code that is used to simulate the model behaviour. The initial versions of the BioUML workbench generated MATLAB code and used the MATLAB ODE suite for simulation (7). The current version of BioUML generates highly optimized Java code and uses its own state-of-the-art simulation engines, which have been developed over the last 14 years. Thanks to the optimized simulation engines, BioUML is the only tool that can pass the SBML (Systems Biology Markup Language) semantic test suite (<http://sbml.org/Facilities/Database/Submission/Details/108>)—a comprehensive set of tests to verify the correctness of simulation engines for complex biological systems (8).

The paradigm of systems biology implies not only computational modelling but also the so-called ‘dry-wet-dry’ cycle composed of theory and computational modelling, which proposes specific testable hypotheses about a bio-

*To whom correspondence should be addressed. Tel: +7 383 363 6829; Email: fedor@biouml.org

logical system, followed by experimental validation and further refinement of the computational model or theory using the newly acquired quantitative description of cells or cell processes (9). Since the objective of such modelling is to describe, as fully as possible, the entire set of interactions in a biological system, high-throughput systems and genome-wide experimental techniques (so-called ‘omics’ data—transcriptomics, metabolomics, proteomics, etc.) are most suitable for validation of such system models. Such high-throughput techniques are mainly used to collect quantitative data for the construction and validation of models. Thus, the second main direction of BioUML development was the processing and analysis of omics data.

To process and analyse omics and other biomedical data, we integrated R/Bioconductor (10) and Galaxy (11) into the BioUML platform and developed 300+ methods for data analysis. To support the concept of reproducible research, the BioUML platform is equipped with a workflow engine that helps to place various methods for analysis into consequent chains/pipelines that can repeatedly perform the same sequence of analytical steps with new input data. BioUML provides a powerful web editor to visually construct such workflows and an engine for the execution of workflows on a server or computer cloud.

For the visualization of omics data, the BioUML platform provides a diagram viewer/editor and a genome browser. Data from omics experiments (transcriptomics, proteomics, metabolomics) can be mapped on different biological pathways and visualized by highlighting corresponding nodes on the diagram.

Thus, BioUML can perform two steps of the system biology ‘dry-wet-dry’ cycle—the modelling and analysis of omics data. The main BioUML vision is to provide a computational platform for building virtual cells, virtual physiological humans and virtual patients.

Currently, scientists from different countries use the BioUML platform for the collaborative reproducible analysis of biomedical data, pathway visualization and genome browsing.

ARCHITECTURE OVERVIEW

Meta-model

The meta-model is the core of the BioUML platform. It provides an abstract layer (compartmentalized attributed graph) for the comprehensive formal description of a wide range of biological systems and other complex systems. The content of biological pathway databases (e.g. Reactome, PantherDB), SBML models, biological pathways in BioPAX format (12), as well as workflows, can be expressed in terms of the meta-model. This formal description can be used both for a visual representation of the structure of biological systems and for automated code generation to simulate the model behaviour.

The meta-model describes a system as three interconnected parts: (i) graph structure: the system structure is described as a compartmentalized graph; (ii) database level: each graph element can contain a reference to an object in biological databases; and (iii) mathematical (executable) model—an y graph element can be associated with an ele-

ment of a mathematical model or an analysis method (e.g. for workflows).

The meta-model structure is problem-domain neutral, so it is used to describe biological models, as well as executable workflows, for the analysis of biomedical data.

Plug-in based architecture

BioUML is based on plug-in architecture (Open Services Gateway Initiative; OSGi) that enables extension of the platform functionality by the addition of new plug-ins. The basic components of the plug-in-based architecture are as follows:

- A plug-in is the smallest unit that can be developed and delivered separately in the BioUML platform. Plug-ins are coded in Java. A typical plug-in consists of Java code in a JAR library, several read-only files and other resources, such as images, message catalogues and native code libraries. A plug-in is described in an XML manifest file called `plugin.xml`. The parsed contents of plug-in manifest files are made available programmatically through a plug-in registry API provided by the Eclipse runtime.
- Extension points are well-defined function points in the system where other plug-ins can contribute functionality.
- An extension is a specific contribution to an extension point. Plug-ins can define their own extension points so that other plug-ins can integrate tightly with them.

Currently, the BioUML platform includes 120+ plug-ins (<http://wiki.biouml.org/index.php/Category:Plugins>) and 36 extension points (http://wiki.biouml.org/index.php/Category:Extension_points).

USER INTERFACE

The BioUML user interface (Figure 1) and architecture were inspired by the Eclipse platform (<https://www.eclipse.org/ide>). The web interface of BioUML is implemented as a single page application and comprises the following main parts:

- **Repository pane:** This pane contains three tabs—database navigation, user data and available methods.
 - **Databases:** On the top level, this contains a list of available biological databases. Each database has its own structure. Usually, it consists of:
 - **Data:** Collections of biological objects (genes, proteins, chemical substances, reactions, etc.)
 - **Diagrams:** Diagrams or models of biological pathways.
 - **Data:** This contains user data organized into projects. The user can create their own project, import omics and other data, analyse the data and invite colleagues for collaborative analyses of corresponding data. All projects (their own and those where the user was invited) are shown in the ‘Collaboration’ folder. A number of projects were created to demonstrate the main possibilities of the BioUML platform.
 - **Analyses:** A set of analyses and workflows. The set is divided into four sections:

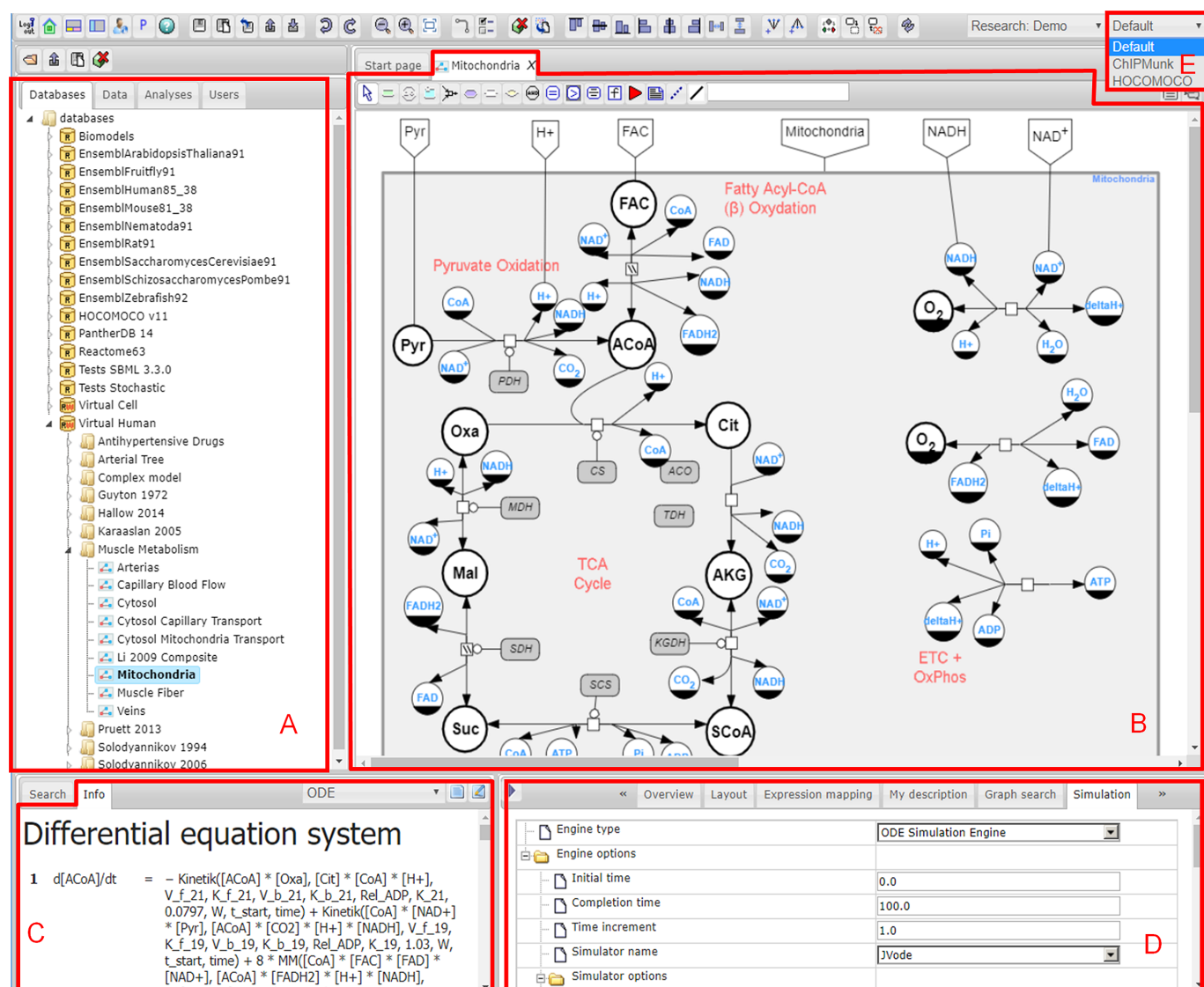


Figure 1. The BioUML web interface consists of: (A) repository pane, (B) document pane, (C) info pane, (D) viewer/editor pane, (E) perspective selector.

- **Galaxy:** Methods of analyses that are available from the Galaxy platform installed on the same server or cloud.
- **Methods:** Analysis methods implemented in Java within the BioUML platform. Currently, it contains 300+ methods grouped into 15 categories. Each method has a detailed description and its own page in the BioUML wiki (e.g. [http://wiki.biouml.org/index.php/Cluster_analysis_by_K-means_\(analysis\)](http://wiki.biouml.org/index.php/Cluster_analysis_by_K-means_(analysis))).
- **JavaScript:** JavaScript API for access to analysis methods.
- **Workflows:** Ready workflows for the analysis of omics data.
- **Document pane:** This is the main working area in the BioUML user interface where tabs with individual documents are opened. There are several document types that provide different functionalities. The main document types are diagram (model), workflow, analysis method and genome browser.
- **Viewer/editor pane:** Each document has a set of tabs (viewers and editors) to work with the current type of documents. Thus, a model document has the following tabs: (i) overview—a small diagram view for navigation; (ii) description; (iii) parameter; (iv) variables—used to display and edit the model parameters and values; (v) simulation—allowing the user to select and start a simulation engine and to configure the simulation parameters; (vi) plot—displays the simulation results; (vii) layout—used to select the graph layout algorithm and apply it to the diagram; (viii) expression mapping—to map and display omics data on the diagram.
- **Info pane:** This shows a description of the selected object in the repository or in the diagram pane. These descriptions are generated using templates and can be opened in a separate tab in the web browser. The user

can select a template if several templates are available. For example, for a diagram (model), the following templates are available: (i) reactions—a list of reactions from the model; (ii) parameters—a list of parameters and their values and measurement units; (iii) variables—a list of variables and their initial values and measurement units; (iv) ODE—a system of differential equations generated from the model; (v) overview—includes all of the information mentioned above.

- **Search pane:** This allows the user to specify criteria for searching information in any database selected in the repository. For this purpose, information of the installed databases is indexed using Apache Lucene (<http://lucene.apache.org>).
- **Perspectives:** This facilitates user concentration on a specific task or a specific database. We have developed several predefined perspectives that display only those web interface elements that are necessary for the specific task.

MAIN FEATURES

Systems biology

BioUML supports the main worldwide standards used in systems biology:

- **SBML:** Systems Biology Markup Language (13) serves for the formal description of mathematical models. BioUML supports all versions of SBML, from 11v2 to the latest 13v2, including the extension packages ‘fbc’ (14) and ‘comp’ (15).
- **SBGN:** Systems Biology Graphic Notation (6) is used for the visual description of model elements (complexes, compartments, molecule types, reactions, etc.). BioUML completely supports SBGN Process Description diagrams and uses them to visually represent SBML models. BioUML also supports the XML markup language SBGN-ML (<https://github.com/sbgn/sbgn/wiki/SBGN-ML>), which facilitates the exchange of SBGN diagrams between tools.
- **Antimony:** Human-readable text format that supports most of the SBML features (16). In BioUML, it is automatically processed into SBML diagrams in SBGN notation. BioUML supports import and export into the antimony format.
- **SedML:** Simulation Experiment Description Markup Language (17) describes model simulation steps and facilitates the reproducibility of simulation experiments. In BioUML, it is translated into workflows, which allows for the analysis and simulation of mathematical models and bioinformatics data.
- Many models, however, require some features that are missing from the above-mentioned standards. In these cases, the SBML standard provides extension mechanisms via the <notes> and <annotation> XML elements. Using these extensions, BioUML stores all additional information about the models (e.g. diagram view attributes and layout).
- SBGN was developed independently from SBML, so it does not define visual syntaxes for events, functions, as-

signments and other mathematical elements. To solve this problem, we have extended the SBGN process diagrams with additional glyphs to represent and use them in our own notations. Detailed information about the types of models and their visual representations can be found at http://wiki.biouml.org/index.php/Diagram_type.

- **Simulation engine:** BioUML automatically generates program code that is used to simulate the behaviour of the analysed model. Currently, BioUML generates highly optimized Java code and uses its own state-of-the-art simulation engines. For each diagram, it provides a list of available engines. For example, network of reactions can be simulated as a system of ODEs or as a Gillespie-type stochastic model. The selected simulation engine provides a list of available solvers. Available ODE solvers include JVODE, which is a package CVODE ported from C to Java and developed at the Lawrence Livermore National Laboratory (18). It utilizes the multistep Adams-Moulton method and the backward differential algorithm, RADAU5 solver (19), as well as classic algorithms (Euler, Dormand-Prince (20)). The stochastic simulation engine provides the exact methods, Gillespie (21) and Gibson-Bruck (22), as well as approximation methods.

Diagram transformation into a simulatable state by the selected simulation engine is a prerequisite for simulation. Thus, a hierarchical diagram may be transformed to an ordinary ‘flat’ diagram with reactions and entities. An agent-based diagram may be partially flattened, where all subdiagrams of the same type may be transformed into one combined agent.

There are several other simpler preprocessors. For example, SBML constraints are transformed into discrete events, thereby halting simulation when the constraint is violated. Additionally, fast reactions are transformed into algebraic equations, and Boolean expressions are transformed into numeric expressions, etc.

Other simulation engines are:

- **Hemodynamics:** specifically tailored to solve PDE problems describing blood flow in arteries.
- **Population:** solves NLME problems using the R library.
- **Dynamic FBA:** dynamically runs Flux Balance Analysis simultaneously with ODE simulation.

Modular modelling

In a modular approach, the investigated system is viewed as a set of interconnected subsystems. Each subsystem can be considered and simulated independently. Integration of these models (or modules) results in a more complex model of the whole system. Modules may leverage different mathematical formalisms and scales. They can be created, validated and improved independently and may be viewed as replaceable parts. Modules provide explicit interfaces through which they can be connected without exposing their inner structure to the user. We consider modules as mathematical models; their interfaces are variables and constant parameters. For example, the value of a variable in one model may be constant, while in another model it changes dynamically. Numerical calculations are performed in two ways:

- (i) Flattening: A modular model may be transformed into a non-modular model by aggregating all elements of all modules with automatic resolving of established connections between variables (23).
- (ii) Agent-based simulation: Each module is simulated independently with its own simulator and formalism. The scheduler coordinates their interactions by sending and receiving numerical values of the connected variables (24).

Parameter estimation

BioUML provides several stochastic and deterministic global optimization methods (25), including a stochastic ranking evolution strategy (26), particle swarm optimization (27), cellular genetic algorithms (28) and others. We have achieved a significant acceleration of these methods using concurrent computing. Algorithms can use experimental data in time-course or steady state forms, with exact or relative values. BioUML also supports multi-experiment parameter estimation. A detailed comparison with other software can be found in (25).

Model analysis

We have implemented a number of methods for model analysis and reduction, including:

- Identifiability analysis infers how well the model parameters are approximated by the amount and quality of experimental data (29,30).
- Search for linear, monomolecular and pseudo-monomolecular reactions (31).
- Quasi-steady state analysis (32).
- Sensitivity analysis of the model steady state (33).
- Metabolic control analysis quantifies how fluxes and species concentrations depend on the system parameters (34).
- Stoichiometric analysis derives linear relationships between flux rates and reactant concentration derivatives (31).
- Mass conservation analysis decomposes a stoichiometric matrix into the product of its linearly independent rows and a link matrix (35).

BIOMEDICAL DATA ANALYSES

For processing and analysis of omics and other biomedical data, we have integrated the best platforms in the respective fields—R/Bioconductor (36) and Galaxy—into the BioUML platform and developed 300+ of our own analysis methods (<http://wiki.biouml.org/index.php/Category:Analyses>).

- **Integration with R.** BioUML has bidirectional integration with R. R scripts can be used within BioUML in four ways: (i) The user can create, edit and execute R scripts in the BioUML document pane. The editor supports syntax highlighting; (ii) The ‘Script’ pane allows the user to input and execute R commands; (iii) R scripts can be building blocks of a BioUML workflow; and (iv) There are a

number of Java analysis tools that provide a convenient interface to configure the analysis parameters, with subsequent generation of the corresponding R script. To execute an R script, the BioUML server calls R. Text output is shown in the ‘Output’ tab. Graphical results (plots, dendrograms, etc.) are shown on separate pages.

To gain access to the BioUML server from inside R, we have developed the `rbiouml` package (<https://cran.r-project.org/package=rbiouml>). The package contains functions to acquire data from the BioUML repository, import/export the data, start analyses and workflows and manage the execution queue.

- **JavaScript API.** The user can use JavaScript (document, console, building block on workflow) similar to R scripts. API provides functions to acquire data from the BioUML repository, import/export the data, start analysis tools and workflows, and provides detailed access to complex BioUML objects (e.g. models). In contrast with R scripts, JavaScript is executed inside the BioUML server.
- **Integration with Galaxy.** The Galaxy platform provides explicit descriptions (Galaxy tool XML file) of parameters for thousands of biological tools, mainly command line tools. BioUML extends the Galaxy tool configuration syntax that allows a closer interaction between the Galaxy and BioUML systems (http://wiki.biouml.org/index.php/Creating_Galaxy_tool).
- BioUML can read these XML files and generate forms where the user can specify values for corresponding parameters of the tools integrated in Galaxy.
- **Workflows.** For reproducible research, analysis tools can be joined into workflows. BioUML provides a powerful editor to visually construct workflows, and the engine for workflow execution is located on a server or cloud.

BioUML workflows can include the following component types:

- Analysis method: Method for analyses with specified inputs/outputs and parameters. It can be a BioUML method, Galaxy tool or Java wrapper for R functions.
- Analysis script: R script or JavaScript code, R methods.
- Analyses parameter: Subset of parameters that the user should specify to start the workflow.
- Analyses expression: Used to set and connect the input and output analysis parameters in the workflow.
- Cycle: Subset of workflow steps that will execute repeatedly. Cycles can iterate over the elements of folder, over table columns, over ranges of integers and over arrays of elements. See <http://wiki.biouml.org/index.php/Workflow> for more details.

PATHWAY VISUALIZATION

The BioUML diagram editor/viewer can be used not only for visual modelling but also for the visualization of different biological pathways. For this purpose, the BioUML server contains the following databases: Reactome (37), PantherDB (38) and Biomodels (<https://www.ebi.ac.uk/biomodels/>). One can load their own pathways in the

following formats: BioPAX, Antimony, SBGN-ML, SBML and Cytoscape CX (39).

BioUML utilizes several algorithms for the automatic layout of visual diagrams, including Hierarchical, Force-directed, Greedy and Grid layouts (40).

Data from omics experiments (transcriptomics, proteomics, metabolomics) can be mapped for different biological pathways and visualized by highlighting corresponding nodes on the diagram (http://wiki.biouml.org/index.php/Expression_mapping).

Integrated genome browser

BioUML provides a fully integrated genome browser (41) that supports most of the features available in other modern genome browsers and comprises a comprehensive set of visualization tools for data processing results, which is extensively used to visualize information from a GTRD database (42).

Collaborative reproducible research

User data (tables, diagrams, etc.) in BioUML are organized into projects. The administrator (creator) of the project can invite other users to participate in the project and manage their permissions. The user registration and management of access rights are performed via a central authentication and authorization system (<https://bio-store.org>). All user actions in a project, including performed analyses and scripts, are tracked in the project journal.

BioUML provides a collaborative editing functionality. Numerical models, pathways and workflows can be simultaneously modified by several researchers, and changes are instantly reflected on the screens of all users, while an embedded chat function facilitates user coordination and collaboration. The system also supports revision control and the possibility to revert to previous versions.

USE CASES

From virtual cell to virtual patient

The BioUML vision is to provide a computational platform to build virtual cells, virtual physiological humans and virtual patients. We have created two databases on the BioUML server that demonstrate our work in this direction using the BioUML platform.

The **Virtual cell** database includes three projects:

- (i) The modular model of apoptosis (23) is the most detailed modern model of apoptosis. The model is split into 13 modules that comprise 280 species (proteins, their complexes, modifications such as different forms of the same molecule, and transformations e.g. phosphorylation) and 372 reactions, applying mass action, as well as Michaelis–Menten kinetics, with 459 parameters.
- (ii) CD95 and NF- κ B signalling pathways (43): When identifying parameters on the basis of experimental data for human cell lines, we were faced with the problem of model overfitting. To solve this problem, we used the technique of model reduction, which allowed

us to obtain a valid set of parameters supported by a sufficient amount of experimental data for modules related to the CD95 and NF- κ B signalling pathways.

- (iii) Complex model of *Mycoplasma genitalium* cell (44): The model consists of 28 submodels utilizing different mathematical formalisms (ODE, stochastic, FBA). Originally, this model was implemented in MATLAB. In 2016, several research groups tried to recreate this model using SBML and SBGN standards (45). Only two submodels were completely finished—Cytokinesis and FtsZ Polymerization.

The **Virtual human** database includes a number of modular models that describe human physiology, including a classic model of blood circulation (46), a model of heart pumping and blood flow (47), a comprehensive model of blood flow through 55 of the largest arteries in the human body (48) and models with a focus on the regulation of blood volume (including kidney) (49,50).

Antihypertensive drugs: This is a database of pharmacokinetic (PK) and pharmacodynamic (PD) models of antihypertensive drugs from different drug groups, including aliskiren, losartan, amlodipine, enalapril, bisoprolol and hydrochlorothiazide.

Complex model: This database combines physiological models with PK/PD models to build so-called ‘virtual patients’. These can be created in diverse forms using different parts of human physiological models, with different focuses on subsystems depending on the research objectives.

Virtual muscle (51): This is a detailed kinetic model describing both the facilitated and passive transport of metabolites between muscle tissues and blood vessels and metabolic processes in cellular compartments (cytosol and mitochondria). We have rebuilt this model as a modular model that became an example of a multilevel model, taking into account cellular compartments and tissue organization.

GTRD database

The GTRD database demonstrates how the BioUML platform can be used to build a web interface for access to a database. We have developed a special GTRD perspective (42,52) that provides browsing, information display, advanced search possibilities, and integration of the genome browser and information from the Ensembl database (gene structures, repeats, etc.) to visualize the GTRD data.

Workflows as a cookbook for the analysis of omics data

Each workflow can be considered as a ready recipe for the specific analysis of corresponding omics data. A scientist needs only to import data, select the appropriate recipe, specify input/output data and press the ‘Run’ button. The platform will automatically analyse the data. This was a key idea of a geneXplain platform (<http://genexplain.com/genexplain-platform/>) (53) that now provides hundreds of workflows for the analysis of different types of omics data (microarrays, transcriptomics, proteomics, metabolomics, etc.). The geneXplain platform is a branch of the BioUML tree, with the focus on commercial application. It includes

such commercial databases as TRANSFAC[®] (transcription factors and their binding sites in a genome; 54), TRANSPATH[®] (signal transduction network in eukaryotic cells; 55) and HumanPSD[®] (disease biomarkers, drugs and clinical trials; 56). The geneXplain platform contains several of its own sophisticated methods for promoter and pathway analysis, such as Match[™] (57) for the identification of transcription factor binding sites, CMA (Composite Module Analysis; 58) for the identification of composite regulatory modules in promoters and enhancers, tools for finding master regulators (59) in networks and other tools.

Recently, a new tool, Genome Enhancer (<http://my-genome-enhancer.com/>), has been developed based on the BioUML platform. Genome Enhancer is a tool for the fully automated analysis of multi-omics data. Depending on a user's data, the platform automatically generates a corresponding workflow, executes the full analysis and presents the results as a well-structured detailed research article.

DISCUSSION

The BioUML platform spans a comprehensive range of capabilities, including access to biological databases, powerful tools for systems biology (visual modelling, simulation, parameters fitting and analyses), a genome browser, scripts (R, JavaScript) and workflows for a diverse array of biomedical data analysis. There is a range of other software platforms that provide similar capabilities for data analysis and modelling with specific extensions for systems biology. The most prominent are:

- **R Studio** provides a web interface and R/Bioconductor provides hundreds of packages for biomedical data analysis.
- **MATLAB** has several packages for biomedical data analysis and systems biology, including SimBiology and IQM Tools (<https://iqmtools.intiquan.com/>), formerly Systems Biology Toolbox). SimuLink provides a powerful tool for the visual development of modular models. For instance, the comprehensive complex model of the bacterial cell, *Mycoplasma genitalium* (44) was created using MATLAB.
- **Jupyter notebook** (60) is widely used interactive computing environment across dozens of programming languages (Python, R, Julia and Scala).

Comprehensive comparisons of the BioUML platform with the above platforms, as well as comparisons with specialized tools for systems biology (e.g. CellDesigner (61), Tellurium (62), COPASI (63), iBioSim (64)), pathway visualization and analyses (e.g. Cytoscape (39)) and workflow platforms (Galaxy, Taverna (65)) are available at http://wiki.biouml.org/index.php/Tools_Comparison.

In general, the BioUML platform has the following advantages:

- A state-of-the-art simulation engine that supports visual modelling using different approaches. As mentioned above, BioUML is the only platform that can pass the SBML semantic test suite, including hierarchical models.

- It provides capabilities for both steps of the systems biology 'dry-wet-dry' cycle—the modelling and analysis of omics data.
- The platform can provide a perspective mechanism to facilitate a user's focus on the tasks and databases they are working with.

ACKNOWLEDGEMENTS

The BioUML team would like to thank all former developers, especially Dr Tagir Valeev and Nikita Tolstykh, for their contribution to the development of the BioUML platform.

FUNDING

Russian Science Foundation [19-14-00295]. Funding for open access charge: Russian Science Foundation [19-14-00295].

Conflict of interest statement. None declared.

REFERENCES

1. Kolpakov, F.A. (2002) BioUML – framework for visual modeling and simulation of biological systems. *Proc. Int. Conf. Bioinf. Genome Regul. Struct. (BGRS'2002)*, Novosibirsk, **2**, 128–131.
2. Kolpakov, F.A., Ananko, E.A., Kolesov, G.B. and Kolchanov, N.A. (1998) GeneNet: a gene network database and its automated visualization. *Bioinformatics*, **14**, 529–537.
3. Kolpakov, F.A. and Ananko, E.A. (1999) Interactive data input into the GeneNet database. *Bioinformatics*, **15**, 713–714.
4. Kolpakov, F., Sharipov, R., Cheremushkina, E. and Kalashnikova, E. (2006) Biopath – a new approach to formalized description and simulation of biological systems. *Proc. Int. Conf. Bioinf. Genome Regul. Struct. (BGRS'2006)*, Novosibirsk, **3**, 96–100.
5. Kolpakov, F., Poroikov, V., Sharipov, R., Kondrakhin, Y., Zakharov, A., Lagunin, A., Milanese, L. and Kel, A. (2007) CYCLONET - an integrated database on cell cycle regulation and carcinogenesis. *Nucleic Acids Res.*, **35**, D550–D556.
6. Le Novère, N., Hucka, M., Mi, H., Moodie, S., Schreiber, F., Sorokin, A., Demir, E., Wegner, K., Aladjem, M.I., Wimalaratne, S.M. et al. (2009) The Systems biology graphical notation. *Nature Biotechnol.*, **27**, 735–741.
7. Shampine, L.F. and Reichelt, M.W. (1997) The MATLAB ODE Suite. *SIAM J. Sci. Comput.*, **18**, 1–22.
8. Hucka, M., Smith, L., Bergmann, F. and Keating, S.M. (2017) *SBML Test Suite release 3.3.0*. <https://zenodo.org/record/1112521#.XNzsIJwxMxE>.
9. Kholodenko, B.N. and Sauro, H.M. (2005) Mechanistic and modular approaches to modeling and inference of cellular regulatory networks. In: Alberghina, L. and Westerhoff, H.V. (eds). *Systems Biology: Definitions and Perspectives. Topics in Current Genetics*. Springer-Verlag, Berlin, Vol. **13**, pp. 357–451.
10. Ramos, M., Schiffer, L., Re, A., Azhar, R., Basunia, A., Rodriguez, C., Chan, T., Chapman, P., Davis, S.R., Gomez-Cabrero, D. et al. (2017) Software for the integration of multiomics experiments in bioconductor. *Cancer Res.*, **77**, e39–e42.
11. Afgan, E., Baker, D., Batut, B., van den Beek, M., Bouvier, D., Cech, M., Chilton, J., Clements, D., Coraor, N., Grünig, B.A. et al. (2018) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic Acids Res.*, **46**, W537–W544.
12. Demir, E., Cary, M.P., Paley, S., Fukuda, K., Lemer, C., Vastrik, I., Wu, G., D'Eustachio, P., Schaefer, C., Luciano, J. et al. (2010) The BioPAX community standard for pathway data sharing. *Nat. Biotechnol.*, **28**, 935–942.
13. Hucka, M., Finney, A., Sauro, H.M., Bolouri, H., Doyle, J.C., Kitano, H., Arkin, A.P., Bornstein, B.J., Bray, D., Cornish-Bowden, A. et al. (2003) The Systems Biology Markup Language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**, 524–531.

14. Olivier, B. and Bergmann, F. (2018) SBML level 3 package: flux balance constraints version 2. *J. Integr. Bioinform.*, **15**, 20170082.
15. Smith, L.P., Hucka, M., Hoops, S., Finney, A., Ginkel, M., Myers, C.J., Moraru, I. and Liebermeister, W. (2015) SBML Level 3 package: hierarchical model composition, version 1 release 3. *J. Integr. Bioinform.*, **12**, 268.
16. Smith, L.P., Bergmann, F.T., Chandran, D. and Sauro, M.H. (2009) Antimony: a modular model definition language. *Bioinformatics*, **25**, 2452–2454.
17. Waltemath, D., Adams, R., Bergmann, F.T., Hucka, M., Kolpakov, F., Miller, A.K., Moraru, I.I., Nickerson, D., Sahle, S., Snoep, J.L. *et al.* (2011) Reproducible computational biology experiments with SED-ML—The Simulation Experiment Description Markup Language. *BMC Syst Biol.*, **5**, 198.
18. Hindmarsh, A.C., Brown, P.N., Grant, K.E., Lee, S.L., Serban, R., Shumaker, D.E. and Woodward, C.S. (2005) SUNDIALS: suite of nonlinear and differential/algebraic equation solvers. *ACM Trans. Math. Soft.*, **31**, 363–396.
19. Hairer, E. and Wanner, G. (1996) Solving Ordinary Differential Equations II. Second Edition. *Stiff and Differential-Algebraic Problems. Springer Series in Computational Mathematics 14*. Springer-Verlag, Berlin, Heidelberg.
20. Dormand, J.R. and Prince, P.J. (1980) A family of embedded Runge-Kutta formulae. *J. Comput. Appl. Math.*, **6**, 19–26.
21. Gillespie, D.T. (2007) Stochastic simulation of chemical kinetics. *Annu. Rev. Phys. Chem.*, **58**, 35–55.
22. Gibson, M.A. and Bruck, J. (2000) Efficient exact stochastic simulation of chemical systems with many species and many channels. *J. Phys. Chem. A*, **104**, 1876–1889.
23. Kutumova, E.O., Kiselev, I.N., Sharipov, R.N., Lavrik, I.N. and Kolpakov, F.A. (2012) A modular model of the apoptosis machinery. *Adv. Experim. Med. Biol.*, **736**, 235–245.
24. Kiselev, I.N., Semisalov, B.V., Biberdorf, E.A., Sharipov, R.N., Blokhin, A.M. and Kolpakov, F.A. (2012) Modular modeling of the human cardiovascular system. *Math. Biol. Bioinform.*, **7**, 703–736.
25. Kutumova, E., Ryabova, A., Valeev, T. and Kolpakov, F. (2013) BioUML plug-in for nonlinear parameter estimation using multiple experimental data. *Virt. Biol.*, **1**, 47–58.
26. Runarsson, T.P. and Yao, X. (2000) Stochastic ranking for constrained evolutionary optimization. *IEEE Trans. Evol. Comput.*, **4**, 284–294.
27. Sierra, M.R. and Coello Coello, C.A. (2005) Improving pso-based multi-objective optimization using crowding, mutation and ϵ -dominance. In: Coello, C.A., Hernández, A., Aguirre, A. and Zitzler, E. (eds). *Evolutionary Multi-Criterion Optimization. EMO 2005. Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, Vol. **3410**, pp. 505–519.
28. Nebro, A.J., Durillo, J.J., Luna, F., Dorronsoro, B. and Alba, E. (2009) MOCcell: A cellular genetic algorithm for multiobjective optimization. *Int. J. Intell. Syst.*, **24**, 726–746.
29. Raue, A., Kreutz, C., Maiwald, T., Bachmann, J., Schilling, M., Klingmüller, U. and Timmer, J. (2009) Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics*, **25**, 1923–1929.
30. Raue, A., Becker, V., Klingmüller, U. and Timmer, J. (2010) Identifiability and observability analysis for experimental design in nonlinear dynamical models. *Chaos*, **20**, 045105.
31. Gorban, A.N., Radulescu, O. and Zinovyev, A.Y. (2009) Asymptotology of chemical reaction networks. *Chem. Engineer. Sci.*, **65**, 2310–2324.
32. Choi, J., Yang, K.W., Lee, T.Y. and Lee, S.Y. (2008) New time-scale criteria for model simplification of bio-reaction systems. *BMC Bioinform.*, **9**, 338.
33. Rabitz, H., Kramer, M. and Dacol, D. (1983) Sensitivity analysis in chemical kinetics. *Ann. Rev. Phys. Chem.*, **34**, 419–461.
34. Reder, C. (1988) Metabolic control theory: a structural approach. *J. Theor. Biol.*, **135**, 175–201.
35. Sauro, H.M. and Ingalls, B. (2004) Conservation analysis in biochemical networks: computational issues for software writers. *Biophys. Chem.*, **109**, 1–15.
36. Huber, W., Carey, V.J., Gentleman, R., Anders, S., Carlson, M., Carvalho, B.S., Bravo, H.C., Davis, S., Gatto, L., Girke, T. *et al.* (2015) Orchestrating high-throughput genomic analysis with Bioconductor. *Nat. Methods*, **12**, 115–121.
37. Croft, D., O’Kelly, G., Wu, G., Haw, R., Gillespie, M., Matthews, L., Caudy, M., Garapati, P., Gopinath, G., Jassal, B. *et al.* (2011). Reactome: A database of reactions, pathways and biological processes. *Nucleic Acids Res.*, **39**, D691–D697.
38. Thomas, P.D., Kejariwal, A., Campbell, M.J., Mi, H., Diemer, K., Guo, N., Ladunga, I., Ulitsky-Lazareva, B., Muruganujan, A., Rabkin, S. *et al.* (2003) PANTHER: a browsable database of gene products organized by biological function, using curated protein family and subfamily classification. *Nucleic Acids Res.*, **31**, 334–341.
39. Shanon, P., Markiel, A., Ozier, O., Baliga, N.S., Wang, J.T., Ramage, D., Amin, N., Schwikowski, B. and Ideker, T. (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.*, **13**, 2498–2504.
40. Kaname, K., Masao, N. and Satoru, M. (2008) Fast grid layout algorithm for biological networks with sweep calculation. *Bioinformatics*, **24**, 1433–1441.
41. Valeev, T., Yevshin, I. and Kolpakov, F. (2013) BioUML genome browser. *Virt. Biol.*, **1**, 15–26.
42. Yevshin, I., Sharipov, R., Kolmykov, S., Kondrakhin, Y. and Kolpakov, F. (2019) GTRD: a database on gene transcription regulation-2019 update. *Nucleic Acids Res.*, **47**, D100–D105.
43. Kutumova, E., Zinovyev, A., Sharipov, R. and Kolpakov, F. (2013) Model composition through model reduction: a combined model of CD95 and NF- κ B signaling pathways. *BMC Syst Biol.*, **7**, 13.
44. Karr, J.R., Sanghvi, J.C., Macklin, D.N., Assad-Garcia, N. and Glass, J.I. (2012) A whole-cell computational model predicts phenotype from genotype. *Cell*, **150**, 389–401.
45. Waltemath, D., Karr, J.R., Bergmann, F.T., Chelliah, V. and Hucka, M. (2016) Toward community standards and software for whole-cell modeling. *IEEE Trans. Biomed. Engineer.*, **63**, 2007–2014.
46. Guyton, A.C., Coleman, T.G. and Granger, H.J. (1972) Circulation: overall regulation. *Ann. Rev. Physiol.*, **34**, 13–46.
47. Proshin, A.P. and Solodovnikov, Y. (2010) Identification of the parameters of blood circulation system. *Automat. Remote Control*, **71**, 1629–1637.
48. Biberdorf, E.A., Blokhin, A.M. and Trakhin, Y.L. (2012) Global modeling of the human arterial system. Ivanova, A.L., Markel, A.M., Blokhin, E.V. and Mishchenko (eds). *Circulatory System and Arterial Hypertension: Experimental Investigation, Mathematical and Computer Simulation*. Nova Science Publishers, Inc., NY, pp. 115–142.
49. Karaaslan, F., Denizhan, Y., Kayserilioglu, A. and Ozcan Gulcur, H. (2005) Long-term mathematical model involving renal sympathetic nerve activity, arterial pressure, and sodium excretion. *Annals Biomed. Engineer.*, **33**, 1607–1630.
50. Hallow, K.M., Lo, A., Beh, J., Rodrigo, M., Ermakov, S., Friedman, S., de Leon, H., Sarkar, A., Xiong, Y., Sarangapani, R. *et al.* (2014) A model-based approach to investigating the pathophysiological mechanisms of hypertension and response to antihypertensive therapies: Extending the Guyton model. *Am. J. Physiol. Reg. Integr. Comp. Physiol.*, **306**, R647–R662.
51. Li, Y., Dash, R.K., Kim, J., Saidel, G.M. and Cabrera, M.E. (2009) Role of NADH/NAD⁺ transport activity and glycogen store on skeletal muscle energy metabolism during exercise: *in silico* studies. *Am. J. Physiol. Cell Physiol.*, **296**, 25–46.
52. Yevshin, I., Sharipov, R., Valeev, T., Kel, A. and Kolpakov, F. (2017) GTRD: a database of transcription factor binding sites identified by ChIP-seq experiments. *Nucleic Acids Res.*, **45**, D61–D67.
53. Kolpakov, F., Poroikov, V., Selivanova, G. and Kel, A. (2011) GeneXplain — identification of causal biomarkers and drug targets in personalized cancer pathways. *J. Biomol. Tech.*, **22**, S16.
54. Wingender, E. (2008) The TRANSFAC project as an example of framework technology that supports the analysis of genomic regulation. *Brief. Bioinformatics*, **9**, 326–332.
55. Krull, M., Voss, N., Choi, C., Pistor, S., Potapov, A. and Wingender, E. (2003) TRANSFAC: an integrated database on signal transduction and a tool for array analysis. *Nucleic Acids Res.*, **31**, 97–100.
56. Michael, H., Hogan, J., Kel, A., Kel-Margoulis, O., Schacherer, F., Voss, N. and Wingender, E. (2008) Building a knowledge base for systems pathology. *Brief. Bioinformatics*, **9**, 518–531.
57. Kel, A.E., Gössling, E., Reuter, I., Cheremushkin, E., Kel-Margoulis, O.V. and Wingender, E. (2003) MATCH: A tool for searching transcription factor binding sites in DNA sequences. *Nucleic Acids Res.*, **31**, 3576–3579.

58. Weber, A., Dittrich-Breiholz, O., Schneider, H., Kel, A., Jauregui, R., Wingender, E. and Kracht, M. (2009) Identification of composite promoter modules in inflammation-regulated genes. *Cell. Commun. Signal.*, **7**, A105.
59. Kel, A.E., Stegmaier, P., Valeev, T., Koschmann, J., Poroikov, V., Kel-Margoulis, O.V. and Wingender, E. (2016) Multi-omics “upstream analysis” of regulatory genomic regions helps identifying targets against methotrexate resistance of colon cancer. *EuPA Open Proteom.*, **13**, 1–13.
60. Perkel, J.M. (2018) Why Jupyter is data scientists’ computational notebook of choice. *Nature*, **563**, 145–146.
61. Funahashi, A., Matsuoka, Y., Jouraku, A., Morohashi, M., Kikuchi, N. and Kitano, H. (2008) CellDesigner 3.5: a versatile modeling tool for biochemical networks. *Proc. IEEE*, **96**, 1254–1265.
62. Choi, K., Medley, J.K., König, M., Stocking, K., Smith, L., Gu, S. and Sauro, H.M. (2018) Tellurium: an extensible python-based modeling environment for systems and synthetic biology. *Biosystems*, **171**, 74–79.
63. Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P. and Kummer, U. (2006) COPASI: a complex pathway simulator. *Bioinformatics*, **22**, 3067–3074.
64. Watanabe, L., Nguyen, T., Zhang, M., Zundel, Z., Zhang, Z., Madsen, C., Roehner, N. and Myers, C. (2018) iBioSim 3: a tool for model-based genetic circuit design. *ACS Synth. Biol.*, doi:10.1021/acssynbio.8b00078.
65. Wolstencroft, K., Haines, E., Fellows, D., Williams, A., Withers, D., Owen, S., Soiland-Reyes, S., Dunlop, I., Nenadic, A., Fisher, P. *et al.* (2013) The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. *Nucleic Acids Res.*, **41**, W557–W561.